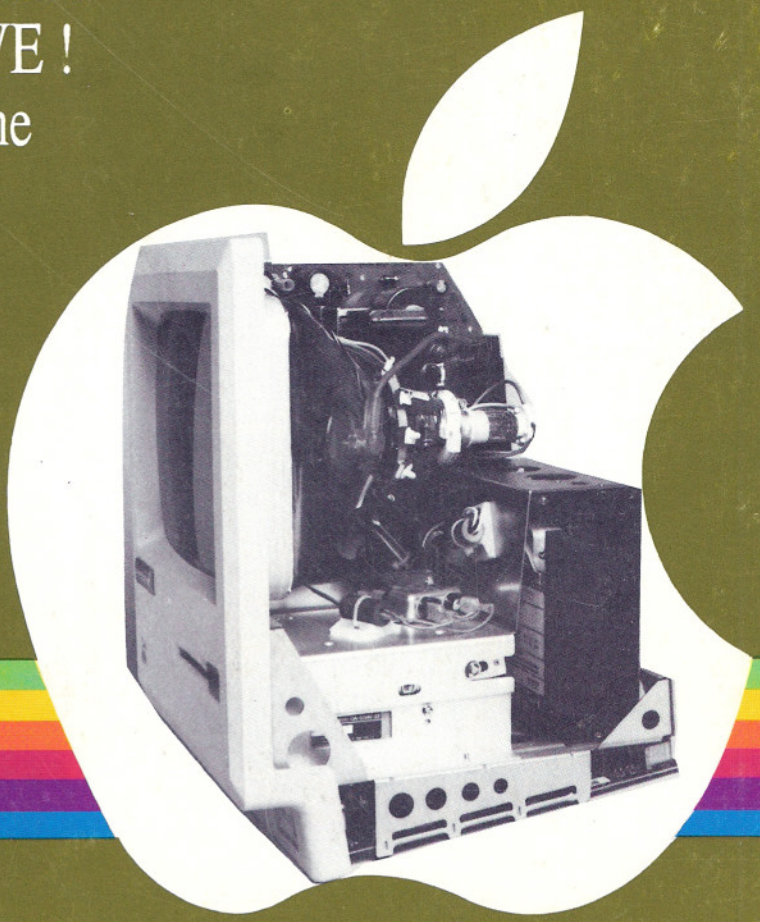
La revue francophone indépendante pour les utilisateurs des Apple ][+, //e, //e+, //c™ et Macintosh™

# pom's

- Haute résolution sur imprimante Apple
- PostScript : deux programmes
- Un disque virtuel sous CP/M
- Snake, un jeu en assembleur
- Macintosh et presse-papiers
- Prudence ? : EXEC SAVE !
- Le Macintosh au téléphone
- L'heure sans carte horloge
- Hard-Copy sur GP500A
- Macintosh : le Bundle
- Analyse multicritère
- Impression Pascal
- Le kit 65C02



NUMERO 24 - PRIX 40 F

M2366-24-40 F

ISSN : 0294-6068

# l'actualité

# les bancs d'essai

# les guides d'achat

# le dossier

# les programmes

# L'ORDINATEUR INDIVIDUEL

## LA RÉFÉRENCE EN MICRO-INFORMATIQUE

***A L'ORDINATEUR INDIVIDUEL**, les rédacteurs, les conseillers techniques, les correspondants à l'étranger, l'équipe entière se mobilise pour vous fournir tous les mois une information complète et de qualité. Le monde de la micro bouge : **L'O.I.** teste pour vous les micros et logiciels qui apparaissent sur le marché. Il vous dit lesquels choisir et pourquoi. Vous êtes déjà équipé et vous souhaitez tirer le maximum de votre machine ? Les spécialistes de **L'O.I.** vous livrent conseils, programmes inédits et astuces d'utilisation. Lisez chaque mois **L'ORDINATEUR INDIVIDUEL.***

**Editorial**
Hervé Thiriez — Page 5

**Un disque virtuel sous CP/M**
Jean-François Rabasse — Page 6

**THRG sur DMP ou ImageWriter**
Jean-Paul Payen — Page 13

**EXEC SAVE**
André Brunel — Page 20

**PostScript**
Christian Piard — Page 21

**Printer**
Philippe Magnien — Page 25

**Datheur**
Bruno Fénart — Page 28

**TOKENisation de chaînes et équations paramétriques**
G. Zwingelstein — Page 35

**Un composeur sur Macintosh**
D. Venot — Page 40

**Le Macintosh et le "presse-papiers"**
Jean-Luc Bazanegue — Page 44

**Le "Bundle" et le "Finder"**
Marianne Sutz — Page 49

**Analyse multicritère**
Serge Cattan — Page 52

**Hard-Copy sur GP500A**
Jean-Marc Roulon — Page 59

**Le kit 65C02**
Pascal Cantot — Page 62

**Snake**
Philippe Krepper — Page 52

**Micro-informations**
Jean-Michel Gourévitch — Page 69

**Bibliographie**
Alexandre Duback — Page 71

**Courrier des Lecteurs**
Page 72

Les annonceurs :
Apple : pages 38 et 39. IEF : page 4. Infomag : page 76. Ordinateur Individuel : page 2. PSI : page 75.



**Editions MEV - 12, rue d'Anjou - 78000 Versailles. Directeur de la publication : Hervé Thiriez**

# I.E.F.

**Le Spécialiste de la Micro *TURBO***

**vous propose**

# Le Macintosh Plus et son *TURBO DISK*

**Caractéristiques Macintosh plus**

- Micro processeur 68000
- 128 K ROM intégrant des fonctions graphiques rapides et la gestion de bureau
- Clavier avec bloc numérique et touches curseur
- **RAM 1 Méga octet** extensible à 4 Mégas
- **Lecteur de disquette 800 K** intégré
- Interface **SCSI** permettant de relier des périphériques puissants
- Système d'exploitation 5.1 avec architecture supérieure et mémoire cache

**Caractéristiques Mac TURBO DISK**

- Disque Dur **20 Méga octets**
- Très haute vitesse d'accès (le plus rapide du marché)
- Haute fiabilité
- Système de Suspension Breveté
- Très faible consommation
- Démarrage automatique sans disquette de boot
- Partitionné en nombre programmable de volumes
- Protection de chaque volume par mot de passe programmable
- Garantie 1 an

**"dernière heure"**
Floppy externe 800 K
**2.900 F** HT
Disque dur interne 20 Méga SCSI
**13.900 F** HT

## A) Vous êtes équipé d'un Macintosh

**IEF** vous transforme votre Macintosh pour seulement :

- **4.500 F** HT (si vous êtes équipe d'un 512 K d'origine Apple)
- **6.500 F** HT (si vous êtes équipé d'un 128 K d'origine ou étendu)
- Pour ces prix, **IEF** vous change la plaque mère, le lecteur de disquettes et le clavier.
- La transformation est garantie 1 an par Apple.
- Ces prix ne sont valables que pendant une durée limitée, réservez dès aujourd'hui votre transformation
- De plus, si vous achetez cette transformation, **IEF vous offre son TURBO DISK au prix de 11.900 F HT**

## B) Vous n'êtes pas encore équipé d'un Macintosh

IEF vous offre **Macintosh Plus** exceptionnellement pour **24.900 F** HT

ou mieux encore la **Promotion spéciale IEF de lancement :**

**1 Macintosh Plus 1 Méga + 1 Mac TURBO DISK 20 Méga 34.900 F** HT (offre limitée)

*IEF propose des conditions spéciales pour les Grands Comptes et les établissements d'enseignement*
*Si vous voulez profiter d'une de ces offres, renvoyez vite le coupon réponse ci-dessous*

**I.E.F.** 217, quai de Stalingrad 92130 ISSY LES MOULINEAUX Tél : (1) 45.57.14.14 Télex : 200210 F

P5 MT

NOM : ..... SOCIETE : .....
ACTIVITE : ..... TEL : .....
ADRESSE : .....
Je suis intéressé par : .....

# *Éditorial*

*En tant qu'éditeur consciencieux, Pom's souffre depuis déjà quelque temps de la richesse des versions de l'Apple II : quand nous testons un programme en vue de sa publication, nous essayons de voir s'il fonctionne sur Apple II Plus, IIe, IIe modifié IIc ; multiplié par DOS et ProDOS, cela donne huit combinaisons... Heureusement que nous avons quelques idées sur ce qui fait qu'un programme a des chances d'être compatible ou non !*

*Jusqu'à présent, ces pénibles recherches nous étaient à peu près évitées avec le Macintosh, qui ne connaissait que les versions 128K et 512Ko. Hélas, l'horizon semble se couvrir de nuages : se rajoute à tout cela le MacPlus, et on entend déjà parler du Mac 512Ko avec lecteur de 800Ko... Dans ce numéro, nous aidons les utilisateurs présents et futurs du MacPlus en leur donnant des indications sur ces irritants problèmes de compatibilité.*

*Chose promise, hélas... dès aujourd'hui nous devons augmenter le prix des disquettes 5,25 pouces mais nous osons ne pas en rougir, vous connaissiez le prix de 55,00 F depuis 1983.*

*Voici dans ces pages - que vous trouverez, nous l'espérons, bien remplies - un beau programme CP/M : un disque virtuel pour les heureux possesseurs de carte 80 colonnes étendue. Si l'on en croit les bulletins de sondage que vous nous avez renvoyés nombreux, vous réserverez bon accueil au jeu "Snake" (ou 'comment cultiver ses réflexes...'). PostScript fait son entrée dans ce numéro ; rares sont ceux qui ont la disposition d'une LaserWriter mais l'aspect "Forth" de la chose ne devrait pas laisser indifférent.*

*Pas moins de 1280 points par ligne : c'est la résolution, souvent inexploitée, de votre imprimante Apple ; il n'en sera plus ainsi, car nous vous proposons un programme qui redonnera à vos courbes mathématiques la finesse que vous souhaitiez obtenir. Toujours au chapitre des imprimantes, c'est à la GP500 que nous donnons un programme de Hard-Copy. En plus des rubriques habituelles, vous découvrirez un utilitaire Pascal, une sauvegarde DOS avec numéros de version, l'art et la manière de 'tokeniser' les chaînes, et enfin Datheur, un utilitaire dont la valeur pédagogique n'échappera pas.*

*Les numéros 22 et 23 proposaient des "Composeurs téléphoniques" pour Apple //+, //e et //c ; il n'y avait aucune raison pour cela s'arrête là... Les utilisateurs du Macintosh trouverons donc dans ce numéro le schéma de cette petite 'interface', ainsi qu'un programme Basic permettant son utilisation immédiate. Dans un autre registre, plus pointu certes, vous trouverez dans le "cahier Mac" deux routines en assembleur propres à faciliter le travail du programmeur. A noter que le respect des notions exposées dans cet article permettrait à nombre de produits commerciaux de ressembler à des programmes 'Macintosh'... Il en va de même pour un autre article, baptisé "Le bundle et le finder".*

*Nous vous donnons rendez-vous à cette grande fête annuelle qu'est Apple Expo ; Pom's vous y accueillera avec plaisir. N'oubliez pas d'emporter ce numéro : vous entrerez à demi-tarif !*

***Hervé Thiriez***

**Ont collaboré à ce numéro :** Alexandre Avrane, Jean-Luc Bazanegue, André Brunel, Pascal Cantot, Serge Cattan, Alexandre Duback, Bruno Fénard, Jean-Michel Gourévitch, Olivier Herz, Philippe Krepper, Philippe Magnien, Gérard Michel, Jean-Paul Payen, Christian Piard, Jean-François Rabasse, Jean-Marc Roulon, Marianne Sutz, Hervé Thiriez, D. Venot, G. Zwingelstein.
**Directeur de la publication, rédacteur en chef :** Hervé Thiriez.
**Rédacteurs :** Alexandre Avrane, Olivier Herz.
**Siège social :** Éditions MEV - 12, rue d'Anjou - 78000 Versailles. Tél. : (1) 39.51.24.43.
**Publicité :** Éditions MEV.
**Diffusion :** N.M.P.P.
**Impression :** Rosay - 47, avenue de Paris - 94300 Vincennes. Tél. : (1) 43.28.18.63.

Pom's est une revue indépendante non rattachée à Apple Computer, Inc. ni à Apple Computer France S.A.R.L. Apple, le logo Apple, Mac et le logo Macintosh sont des marques déposées d'Apple Computer, Inc.

# Un disque virtuel sous CP/M

J-F Rabasse

Le RAMdisk ou disque virtuel, représente le mode d'utilisation le plus souple de cartes d'extension mémoire. Il offre les services d'un disque de capacité réduite mais de coût nul et de vitesse d'accès très élevée.

De tels utilitaires existent sur Apple, pour fonctionner sous DOS 3.3 ou CP/M, proposés par les fabricants d'extensions mémoire (Legend 128 KDE, Titan etc.). Dans le cas particulier de l'extension 64Ko de l'Apple //e (carte 80 colones étendue), un RAMdisk est implanté par ProDOS, d'autres sont disponibles pour le DOS 3.3 (Cf Pom's 14 par exemple). Le programme présenté ici vient compléter la collection en proposant un RAMdisk de 60Ko sous CP/M 2.23, implanté dans la RAM auxiliaire. (On peut ajouter que l'utilisation d'un RAMdisk sous CP/M est très agréable, ce système d'exploitation passant presque autant de temps à faire tourner ses disques qu'à se mettre à la disposition de l'utilisateur.

- La programmation en langage machine avec le cycle Éditeur, Assembleur avec son fichier source, son fichier objet, son fichier listing, puis Debuger et re-Éditeur, est accélérée de façon sensible.
- Le basic Microsoft offre une gestion des fichiers à accès direct très puissante, qui devient très efficace avec les fichiers en RAM auxiliaire.
- Enfin la mise en œuvre de CP/M est moins acrobatique pour ceux qui ne possèdent qu'un seul lecteur).

## Configuration requise

Un Apple //e avec sa carte 80 colonnes étendue, et une carte Z80 avec CP/M 2.23. Attention, Microsoft livre son CP/M en version 44Ko (fonctionnement sur un Apple sans carte langage), reconfigurable en 60Ko avec l'utilitaire CPM60.
Ce programme est interfacé avec le CP/M 60Ko (configuration normale sur un Apple //e.)
D'autre part le programme est prévu pour un système avec 1 ou 2 lecteurs de disquettes maximum. Pour un système à 3 ou 4 drives, quelques petites modifications sont indiquées plus loin.
NB : Il existe depuis peu une extension Z80 CP/M pour l'Apple //c. Ce programme devrait pouvoir être adapté. L'essai n'a pas été fait.

## Création du programme

Avec un assembleur Z80, éditer puis assembler le programme source. Le programme objet est un fichier CP/M .COM, donc l'adresse d'implantation est 0100H (début du TPA).
Sans assembleur, entrer le code objet hexadécimal en utilisant DDT, à partir de l'adresse 0100. (Il est indispensable de vérifier très soigneusement le code frappé. Ce programme manipule les commutateurs des pages mémoire et modifie les routines d'accès disque de CP/M. Des erreurs de frappe peuvent entrainer des effets grandioses à l'exécution. Si vous n'êtes pas sûr de votre virtuosité au clavier, il n'est pas inutile de faire les premiers essais avec une disquette copie !)
Sauvegarder le code par :

```
SAVE 3 RAMdisk.COM
```

## Utilisation

Elle est très simple. Au niveau commande de CP/M, frapper:

```
RAMdisk <return>
```

Le programme s'exécute puis affiche le message

```
M:  en service.....
```

et retourne au CCP.

Le disque virtuel est prêt. Il est repéré par M (Mémoire). On y accède par `M:` . On peut y transférer les fichiers et utilitaires désirés par PIP. Pour des utilisations régulières il est pratique de créer un fichier Batch, comportant l'appel de RAMdisk et tous les transferts souhaités (PIP, ED, DDT, etc.) lancé par une commande SUBMIT.

Une remarque : il peut arriver, en cours de travail, un "plantage" (si si !), qui oblige à rebooter complètement CP/M et donc à devoir réimplanter le RAMdisk. Dans ce cas, le programme détecte la présence du disque virtuel en Ram auxiliaire et affiche le message suivant :

```
Un RAMdisk existe....
faut-il l'annuler  (O/N)?
```

En tapant N, on a alors la possibilité de ne pas réinitialiser la RAM auxiliaire et donc de pouvoir récupérer ses fichiers de travail qui, conformément à la loi de Murphy, n'étaient pas sauvegardés sur une vraie disquette lors du plantage. Attention, ce contrôle consiste à vérifier la présence du code de gestion du RAMdisk. Si un plantage sévère a eu lieu lors de la mise au point d'une routine qui sévit en RAM auxiliaire, l'intégrité
des fichiers n'est pas assurée. Si, en revanche, on souhaite redémarrer sur des bases nouvelles il suffit de répondre O à la question. (On peut aussi taper Y ).

## Fonctionnement

Avant de détailler le fonctionnement de RAMdisk, il

est utile de rappeler le fonctionnement de CP/M et la gestion des disques.
CP/M est constitué de trois grandes parties :

• le CCP (Console Command Processor) qui gère toutes les commandes de l'utilisateur, en provenance du clavier.

• le BDOS (Basic Disk Operating System) qui est l'interface entre les programmes d'application et le système d'exploitation, et qui réalise toutes les fonctions d'entrées/sorties, accès aux fichiers disques, gestion des périphériques. Ces fonctions au nombre de 41 sont documentées dans les ouvrages traitant de CP/M. L'accès au BDOS se fait par l'adresse système 0005, avec le code de la fonction désirée dans le registre C du Z80 ;

• Le BIOS (Basic Input Output System) qui est l'interface entre le BDOS et le calculateur sur lequel est implanté CP/M. Ce BIOS est spécifique à chaque machine, c'est lui qui assure la gestion du "Hard". C'est donc en pratique au niveau du BIOS que se feront toutes les modifications ou adaptations de CP/M.

Le BDOS accède au BIOS par l'intermédiaire d'une table de sauts, qui rassemble les fonctions élémentaires d'entrée-sortie. Cette table a la structure standard suivante :

```
JP  CBOOT   ;initialisation à froid
JP  WBOOT   ;initialisation à chaud
JP  CONSTS  ;état de la console E/S
JP  CONSIN  ;entrée console
JP  CONSOUT ;sortie console
JP  LSTOUT  ;sortie imprimante
JP  PTPUNCH ;sortie perforateur
JP  PTREAD  ;lecture ruban perforé
JP  TRK0    ;remise à 0 tête de lecture disque
JP  SELDRV  ;sélection d'un lecteur de disques
JP  SETTRK  ;choix de la piste
JP  SETSECT ;choix du secteur
JP  SETDMA  ;adresse de transfert
JP  READ    ;lecture d'un secteur
JP  WRITE   ;écriture d'un secteur
```

Sur Apple, sous CP/M 2.23 60Ko, cette table est implantée de 0FA00 à 0FA2C :

```
FA00:  00          NOP
FA01:  00          NOP
FA02:  00          NOP
FA03:  C3 B8 FA    JP    0FAB8H
FA06:  C3 10 FB    JP    0FB10H
FA09:  C3 1A FB    JP    0FB1AH
FA0C:  C3 4D FB    JP    0FB4DH
FA0F:  C3 70 FB    JP    0FB70H
FA12:  C3 7F FB    JP    0FB7FH
FA15:  C3 91 FB    JP    0FB91H
FA18:  C3 75 FE    JP    0FE75H
FA1B:  C3 97 FE    JP    0FE97H
FA1E:  C3 80 FE    JP    0FE80H
FA21:  C3 F4 FB    JP    0FBF4H
FA24:  C3 F9 FB    JP    0FBF9H
FA27:  C3 C6 FE    JP    0FEC6H
FA2A:  C3 CC FE    JP    0FECCH
```

A noter que la fonction ColdBOOT n'est pas implantée puisque le démarrage à froid n'est pas assuré par CP/M et le Z80, mais par la ROM Apple C6 et le 6502.

CP/M gère les disques à partir d'une adaptation de la routine Apple RWTS (implantée dans le Bank 1 à l'adresse $D420 ou 0B420H), et d'un buffer de 256 octets situé en 0F800H. Les routines de lecture/écriture gèrent ce buffer pour adapter les demandes BDOS de secteurs 128 octets, en accès disque par secteurs de 256 octets. La communication des paramètres se fait par des tables. Une table située en 0F3E0-0F3EB (adresses 6502 $3E0-$3EB) contient les paramètres de RWTS : numéros de piste, de secteur, de slot, de drive, code erreur, ancien slot, ancien drive, etc.

Une autre table située en 0FEE8 - 0FEFF, contient tous les paramètres nécessaires au BDOS et au BIOS pour la gestion du disque.

Ceux qui sont indispensables à la bonne marche de RAMdisk sont :

| | |
|---|---|
| 0FEEA | numéro de la piste demandée |
| 0FEEB | numéro du secteur demandé |
| 0FEFA/ 0FEFB | adresse du buffer à lire ou écrire |
| 0FEEF | numéro du lecteur en cours |
| 0F3EA | code erreur au retour de RWTS |

## Modification des routines BIOS

Certaines fonctions n'ont pas a être modifiées. TRK0, SETTRK, SETSECT et SETDMA se bornent à recopier dans la table de paramètres la valeur numérique adéquate.
La fonction SELDSK (implantée de 0FE97 à 0FEC5) est appelée avec, dans le registre C, le numéro du lecteur désiré (0 pour A:, 1 pour B: etc.). Cette routine vérifie que le numéro d'appel est valide (l'adresse 0F3B8 contient le nombre de lecteurs présents, détectés lors du Boot), puis retourne dans le registre HL l'adresse de la table de paramètres du lecteur. Ces tables de paramètres, longues de 16 octets sont situés en 0FA33, 0FA43, 0FA53, 0FA63, pour les lecteurs A:, B:, C: et D:
La modification de SELDSK consiste à tester si le numéro du lecteur appelé correspond au RAMdisk reperé par M: soit un numéro 12. Si oui, on recopie le numéro à l'adresse 0FEEF, et on retourne dans HL l'adresse des paramètres RAMdisk. (cette table a été installée en 0FA53, zone libre sur un système à deux drives). Si un autre lecteur est appelé, on rend la main à la routine SELDSK standard.

Les fonctions READ et WRITE commutent le Bank 1 de la ram Apple (où sont situées toutes les routines d'accès disque) et sautent à la routine de gestion de RWTS implantée de 0B247 à 0B358, point d'entrée lecture en 0B247, point d'entrée écriture en 0B257.
La modification consiste à tester si le numéro du lecteur en cours

(0FEEF) est celui du RAMdisk, et le cas échéant à détourner vers les routines de lecture/écriture RAMdisk. En sortie de ces routines, on copiera dans 0F3EA, un 0 ou un code d'erreur eventuel (codes d'erreurs de RWTS).

### Remarques concernant la table des paramètres disque

Pour chaque lecteur de disques, CP/M utilise une table de 16 octets contenant les adresses de zones de travail. Cette table comporte 8 octets réservés à CP/M, l'adresse d'un buffer de 128 octets pour le catalogue des disques (cette adresse est la même pour tous les lecteurs : 0FEFD), l'adresse des paramètres d'organisation de la disquette, et les adresses de deux autres buffers de 12 et 18 octets respectivement. Le CP/M Apple peut gérer jusqu'à 4 disques A, B, C, D. Microsoft a prévu l'implantation de 4 tables, de 0FA33 à 0FA72, et tous les buffers de travail nécessaires. Si, par exemple, on n'utilise que 2 lecteurs, on peut "emprunter" la table prévue pour le drive C: 0FA53.
La seule modification à apporter concerne l'adresse de la table d'organisation de la disquette (11eme et 12eme octets). Les lecteurs Apple étant tous identiques, une seule table est prévue pour les 4 lecteurs possibles. Cette table, longue de 15 octets, est implantée en 0FA73. Pour le RAMdisk, il est nécessaire de créer une autre table, identique à deux octets près : les 6 et 7èmes qui contiennent la taille du disque sous forme du plus haut numéro de bloc. Avec un RAMdisk de 60Ko, donc 60 blocs, on devra écrire 59 dans cette table soit 3B 00.

La table d'organisation de la disquette peut être placée en 0FA63 toujours si l'on n'utilise que 2 lecteurs. En revanche, sur un Apple équipé de 4 lecteurs, il faudra créer et placer les deux tables de paramètres et tous les buffers nécessaires.

Tables des paramètres lecteurs de CP/M 2.23 60k :

```
FA33   00 00 00 00 00 00 00 00 FD FE 73 FA C5 FF 7D FF
FA43   00 00 00 00 00 00 00 00 FD FE 73 FA D1 FF 8F FF
FA53   00 00 00 00 00 00 00 00 FD FE 73 FA DD FF A1 FF
FA63   00 00 00 00 00 00 00 00 FD FE 73 FA E9 FF B3 FF
```

Table des paramètres d'organisation des disquettes :

```
FA73   20 00 03 07 00 8B 00 2F 00 C0 00 0C 00 03 00
```

## Fonctionnement du RAMdisk

Afin de limiter la place occupée en mémoire centrale, la majorité du code de gestion du RAMdisk est implantée en Ram auxiliaire. Seulement une cinquantaine d'octets sont en mémoire principale. Ils ont été placés de 0F100 à 0F130, zone inutilisée par CP/M car c'est la pile du 6502. Le 6502 n'étant utilisé que pour gérer des routines d'entrées-sorties, il est peut probable qu'il remplisse sa pile si haut.

Le programme comporte une partie préparation de CP/M, et la partie gestion du RAMdisk, qui reste en machine.

Lorsqu'on lance le programme, il commence par comparer les 256 premiers octets du code de gestion avec le contenu de la ram auxiliaire de 0F100 à 0F1FF. En cas d'identité, un pseudo disque est censé exister et la Ram auxiliaire n'est pas réinitialisée. Seul CP/M est reconfiguré.

Le programme se recopie ensuite en Ram auxiliaire et commute la Ram, pour transferer l'exécution en Ram auxiliaire. Ceci permet la mise en place du code de gestion du RAMdisk (0F100 à 0F2C7), et "l'initialisation" de la piste 3 (Catalogue CP/M) en remplissant de codes E5.

Le programme revient ensuite en Ram principale. L'accès au code de gestion RAMdisk est fait en recopiant le début du code (0F100 à 0F10D) en Ram principale. Ce tronçon comporte les "portes" d'entrées-sorties et la commutation de mémoire.

Pour configurer CP/M il reste à installer les trois routines modifiées READ, WRITE et SELDSK, de 0F10E à 0F130, et modifier les adresses du vecteur BIOS afin de pointer ces routines.
Enfin on installe la table des paramètres de la pseudo disquette en 0FA63, et on pointe cette table dans le bloc CP/M réservé au pseudo disque, installé en 0FA53 (à la place d'un lecteur C:).
Le programme affiche un message et rend la main au CCP.

## Code de gestion du RAMdisk

La routine de lecture écriture du pseudo disque comporte quatre phases :

Récupération des paramètres de l'appel disque et transfert de l'exécution en Ram auxiliaire, puis création d'une pile en Ram auxiliaire pour le Z80.

Calcul d'une adresse mémoire correspondant aux numéros de piste et secteur.
Les pistes 0,1 ... 0D sont placées en 00000,01000 .... 0D000, et la piste 0E en 0D000 Bank 1. Le début théorique d'une disquette CP/M étant la piste du catalogue, et celui ci étant placé en piste 3, il faut transformer les numéros de pistes 0F,10,11 appelés par le BDOS en 0,1,2. En cas d'appel d'un numéro supérieur, le programme retourne à CP/M avec le code 040 dans 0F3EA (code I/O Error de RWTS).

Transfert du secteur.
Le transfert est délicat à cause de la commutation des pages de mémoire. Le secteur RAMdisk

peut se trouver dans la Ram 48Ko ou dans la mémoire à Bank commuté, dans le Bank 1 ou le Bank 2. Le buffer CP/M peut être dans la Ram 48Ko ou la mémoire à Bank commuté, ou à cheval sur les deux. (C'est le cas par exemple avec le logiciel ED.COM qui charge ses fichiers en haut du TPA, sous l'adresse 0DC06. On a donc en général un secteur chargé de 0AF86 à 0B006).
Certains transferts doivent se faire en deux temps, avec transit par un secteur temporaire situé en 0F300.
Pour obtenir un fonctionnement très rapide, on a choisi de faire les transferts par bloc complet en utilisant la puissante instruction LDIR du Z80, à l'aide de huit routines de transfert spécifiques à chaque cas, qui assurent le calcul des adresses et les commutateurs de mémoire à actionner.

Le résultat est un code plus long qu'avec une routine unique de transfert-commutation octet par octet, mais le temps d'accès se trouve réduit dans un rapport quatre.
Le transfert proprement dit est effectué par deux routines COPI1 et COPI2 situées l'une dans la mémoire 48Ko l'autre dans la mémoire à Bank. Ces routines sont indépendantes de l'état des commutateurs mémoire puisque tous les paramètres nécessaires - adresses de transfert et commutateurs à actionner - sont passés dans les registres 16 bits HL, BC, DE, HL' ,BC' ,DE' du Z80.

Retour en Ram principale et au BDOS.

## Source 'RAMDISK.TEXT'

```
        TITLE   RAMDISK
;
;*****************************************************
;*                                                   *
;*     DISQUE VIRTUEL SOUS CP/M 2.23                 *
;*          VERSION  CP/M  60k                       *
;*                                                   *
;*          APPLE //e   128 k                        *
;*                                                   *
;*      JF-R            JUN 85                       *
;*****************************************************
;
;COMMUTATEURS DES PAGES DE MEMOIRE
;
RDMAIN  EQU     0E002H  ;RAM PRINCIPALE EN LECTURE
RDAUX   EQU     0E003H  ;RAM AUXILIAIRE EN LECTURE
WRMAIN  EQU     0E004H  ;RAM PRINCIPALE EN ECRITURE
WRAUX   EQU     0E005H  ;RAM AUXILIAIRE EN ECRITURE
BKMAIN  EQU     0E008H  ;RAM A BANC COMMUTE PRINCIPALE
BKAUX   EQU     0E009H  ;RAM A BANC COMMUTE AUXILIAIRE
BK1     EQU     0E08BH  ;BANK 1
BK2     EQU     0E083H  ;BANK 2
;
;
;STOCKAGE DES VARIABLES
;
SAVSP   EQU     0F080H  ;SAUVEGARDE POINTEUR DE PILE
OPER    EQU     SAVSP+2 ;CODE OPERATION:
;                       0 LECTURE, 2 ECRITURE
BUFF    EQU     OPER+1  ;ADRESSE BUFFER CP/M
SECT    EQU     BUFF+2  ;ADRESSE SECTEUR DEMANDE
BANK    EQU     SECT+2  ;CODE BANK DEMANDE
TRK     EQU     BANK+2  ;NO DE PISTE CP/M
SCT     EQU     TRK+1   ;NO DE SECTEUR CP/M
;
;PARAMETRES
;
RAMDSK  EQU     'M'-'A' ;CODE M: POUR LE RAMDISK
TEMP    EQU     0F300H  ;STOCKAGE TEMPORAIRE
STACK   EQU     0F100H  ;NOUVELLE PILE Z80
NBOCT   EQU     80H     ;NB D'OCTETS A TRANSFERER
;
BDOS    EQU     5       ;ENTREE BDOS
;
;=================================================
; PROGRAMME RAMDISK
; ROUTINE DE PREPARATION
;=================================================
;
DEB     EQU     $
;
;
;ROUTINE D'IDENTIFICATION DU RAMDISK
;
        LD      HL,BLF
        LD      DE,BL0
        LD      BC,100H
        LD      (BKAUX),A
IDENT1: LD      A,(DE)
        CPI
        JR      NZ,IDENT2
        JP      PO,IDENT2
        INC     DE
        JR      IDENT1
IDENT2: LD      (BKMAIN),A
        JR      NZ,PREP         ;PAS DE RAMDISK
        LD      C,9
        LD      DE,MSG
        CALL    BDOS            ;MESSAGE DE CONFIRMATION
        LD      C,1
        CALL    BDOS            ;REPONSE?
        LD      HL,CARAC
        LD      BC,4
        CPIR
        JR      Z,PREP          ;PREPARE LA RAM AUX.
        JR      MODIF           ;CONFIG. CP/M
CARAC:  ASC     'OYoy'          ;CARACTERES VALIDES
MSG:    DFB     0DH,0AH
        ASC     'Un Ramdisk existe....'
        DFB     0DH,0AH
        ASC     'faut-il l''annuler  (O/N)?  $'
;
;RECOPIE DU CODE COMPLET EN RAM AUXILIAIRE
;
PREP:   LD      HL,DEB
        LD      DE,DEB
        LD      BC,LPROG
        LD      (WRAUX),A
        LDIR
;
;TRANSFERT DE L'EXECUTION EN RAM AUXILIAIRE
;
        LD      (RDAUX),A
        LD      (BKAUX),A
;
;MISE EN PLACE DU CODE DE TRANSFERT
;
        LD      HL,BL0
        LD      DE,BLF
        LD      BC,LG2
        LDIR
;
;INITIALISATION PISTE 3 (DIRECTORY CP/M)
;
```

```
        LD      HL,3000H
        LD      A,0E5H
        LD      BC,0FFFH
        LD      (HL),A
        LD      DE,3001H
        LDIR
;
;RETOUR EN RAM PRINCIPALE
;
        LD      (BKMAIN),A
        LD      (WRMAIN),A
        LD      (RDMAIN),A
;
;==================================================
; MODIFICATIONS DU CP/M EN RAM PRINCIPALE
;==================================================
MODIF   EQU     $
;
;MISE EN PLACE DU CODE D'ENTREE SORTIE
;
        LD      HL,BL0
        LD      DE,BLF
        LD      BC,LG1
        LDIR
;
;MISE EN PLACE DU CODE DE DETOURNEMENT
;
        LD      HL,DET0
        LD      DE,DETF
        LD      BC,LGDET
        LDIR
;
;MISE EN PLACE DU BLOC PARAMETRES DISQUE
;
        LD      HL,BLPAR
        LD      DE,0FA63H
        LD      BC,LGBL
        LDIR
;
;MODIF TABLE DISQUE
;
        LD      HL,0FA63H
        LD      (0FA5DH),HL
;
;MODIF VECTEURS BIOS
;
        LD      HL,READ
        LD      (0FA28H),HL
        LD      HL,WRITE
        LD      (0FA2BH),HL
        LD      HL,SELDSK
        LD      (0FA1CH),HL
;
        LD      C,9
        LD      DE,MSGOK        ;MESSAGE TERMINE
        JP      BDOS            ;SORTIE BDOS ET RETOUR
;                                 PAR LE CCP.
MSGOK:  DFB     0DH,0AH,0DH,0AH
        DFB     RAMDSK+'A'
        ASC     ':  en service.....'
        DFB     0DH,0AH
        ASC     '$'
;
;==================================================
;    IMPLANTATION DU RAMDISK EN RAM AUXILIAIRE
;==================================================
;
BL0     EQU     $       ;DEBUT DU CODE DE TRANSFERT
BLF     EQU     0F100H  ; ET ADRESSE D'IMPLANTATION
        .PHASE  BLF
;
;
;==================================================
;  ROUTINES D'ACCES:  CE CODE EST COMMUN AUX DEUX
;                PAGES F1, PRINCIPALE ET AUXILIAIRE
;==================================================
;
RD      EQU     $       ;POINT D'ENTREE LECTURE
        XOR     A
        JR      COMMUT
WR      EQU     $       ;POINT D'ENTREE ECRITURE
        LD      A,2
        JR      COMMUT
SORTIE  EQU     $       ;POINT DE RETOUR DES ROUTINES
        LD      (BKMAIN),A
        RET
COMMUT: LD      (BKAUX),A
;
LG1     EQU     $-BLF
DETOUR  EQU     $               ;IMPLANTATION DE LA
;                                ROUTINE DE DETOURNEMENT
;
;LA SUITE S'EFFECTUE EN RAM AUXILIAIRE
;
;
;       RECUPERATION DES PARAMETRES ET CREATION
;       D'UNE NOUVELLE PILE EN RAM AUX.
;
        LD      (OPER),A
        LD      HL,STACK
        LD      (SAVSP),SP
        LD      SP,HL
        LD      HL,(0FEFAH)     ;ADR. DU BUFFER CP/M
        LD      (BUFF),HL
        LD      HL,(0FEEAH)     ;PISTE ET SECT. DEMANDES
        LD      (TRK),HL
        LD      (RDAUX),A
        LD      (WRAUX),A
;
;
;==================================================
;  CALCUL DE L'ADRESSE DU SECTEUR DEMANDE
;    ET TRANSFERT DU BLOC DE 128 OCTETS
;==================================================
;
CALCAD: LD      A,(TRK)
        CP      12H             ;SI >= 12, I/O ERREUR
        JR      NC,ERREUR+1
        CP      0FH             ;SI >= 0F, "REPLIEMENT"
        JR      C,CALC1
        SUB     0FH             ; DU NO DE PISTE
CALC1:  LD      HL,BK2
        CP      0EH             ;SI =0E, EN BANK1
        JR      NZ,CALC2
        LD      HL,BK1
        LD      A,0BH           ;NOUVEAU NO DE PISTE
        LD      (TRK),A
CALC2:  LD      (BANK),HL
;
;ADRESSE= TRK*4096 +SCT*128
;
        RLCA
        RLCA
        RLCA
        RLCA
        LD      D,A
        LD      HL,(TRK)
        XOR     A
        LD      E,A
        RR      H
        RRA
        LD      L,A
        ADD     HL,DE
        LD      (SECT),HL
        CALL    TRANSF          ;TRANSFERT DU SECTEUR
        XOR     A
ERREUR: LD      HL,403EH        ;ERREUR+1= LD A,40H
        LD      (BK2),A
        LD      HL,(SAVSP)      ;RECUPERATION DE LA PILE
        LD      SP,HL
        LD      (RDMAIN),A
        LD      (WRMAIN),A
```

```
        LD      (0F3EAH),A      ;CODE ERREUR RWTS
        JR      SORTIE          ;RETOUR AU BDOS
;
;
;==================================================
;ROUTINE DE COMMUTATION DES PAGES ET COPIE DE BLOC
;       RESIDE EN BANK AUX
;==================================================

COPI1:  LD      (HL),A          ;RAM EN LECTURE
        LD      (DE),A          ;RAM EN ECRITURE
        EXX
        LDIR                    ;TRANSFERT BLOC
        LD      (RDAUX),A       ;REMISE EN LECT.
        LD      (WRAUX),A       ; ET ECRITURE DE LA
        RET                     ; RAM AUXILIAIRE
;
;
;==================================================
;  ROUTINE DE TRANSFERT
;==================================================
;
;DETERMINATION DU TYPE DE TRANSFERT A EFFECTUER
;
TRANSF: LD      A,H             ;OCTET HAUT DE SECT
        CP      0B0H            ;EN RAM BANK ?
        LD      A,(OPER)        ;LECT. OU ECRIT. ?
        ADC     A,0
        RLCA
        LD      C,A
        LD      DE,(BUFF)
        LD      HL,0AFFFH
        LD      A,0F0H          ;SI BUFFER CP/M EN FXXX
        CP      D               ; IL EST EN MAIN48
        JR      NC,TRANS0
        LD      D,1
TRANS0: XOR     A
        SBC     HL,DE
        JR      C,TRANS1        ;BUFFER EN BANK MAIN
        INC     HL
        INC     C
        RL      C
        XOR     A
        CP      H
        JR      NZ,TRANS2       ;BUFFER EN MAIN48
        LD      A,7FH
        SUB     L
        JR      C,TRANS2        ;BUFFER EN MAIN48
        INC     C               ;SINON, BUFFER A CHEVAL
        INC     A               ; SUR B000
        LD      D,H
        LD      E,A
        LD      A,(OPER)
        OR      A
        JR      Z,TRANS2
        PUSH    DE              ;SI ECRITURE, ON EMPILE
        PUSH    HL              ;LES NB D'OCTETS APRES
        JR      TRANS2          ; ET AVANT B000
TRANS1: CCF
        RL      C
TRANS2: XOR     A
        LD      B,A
        RL      C
        LD      HL,ADTR         ;LECTURE DE L'ADRESSE DE
        ADD     HL,BC           ;LA ROUTINE DE TRANSFERT
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        EX      DE,HL
        JP      (HL)            ;ET EXECUTION
;
;TABLE DES ADRESSES DES ROUTINES DE TRANSFERT
;
ADTR:   DW      TR1,TR1,TR2,TR1
        DW      TR3,TR3,TR3,TR3
        DW      TR4,TR4,TR5,TR6
        DW      TR7,TR7,TR5,TR8
;
;
TR1:    LD      HL,(SECT)
        LD      DE,TEMP
        LD      BC,NBOCT
        EXX
        CALL    MOVE5
        LD      HL,TEMP
        LD      DE,(BUFF)
        LD      BC,NBOCT
        EXX
        JP      MOVE4
;
TR2:    LD      HL,(SECT)
        LD      DE,(BUFF)
        LD      BC,NBOCT
        EXX
        JR      MOVE2
;
TR3:    LD      HL,(SECT)
        LD      DE,(BUFF)
        LD      BC,NBOCT
        EXX
        JP      MOVE4
;
TR4:    LD      HL,(BUFF)
        LD      DE,TEMP
        LD      BC,NBOCT
        EXX
        CALL    MOVE3
        LD      HL,TEMP
        LD      DE,(SECT)
        LD      BC,NBOCT
        EXX
        JR      MOVE5
;
TR5:    LD      HL,(BUFF)
        LD      DE,(SECT)
        LD      BC,NBOCT
        EXX
        JR      MOVE1
;
TR6:    LD      HL,(BUFF)
        LD      DE,TEMP
        POP     BC
        EXX
        CALL    MOVE1
        POP     BC
        EXX
        CALL    MOVE3
        LD      HL,TEMP
        LD      DE,(SECT)
        LD      BC,NBOCT
        EXX
        JR      MOVE5
;
TR7:    LD      HL,(BUFF)
        LD      DE,(SECT)
        LD      BC,NBOCT
        EXX
        JR      MOVE3
;
TR8:    LD      HL,(BUFF)
        LD      DE,(SECT)
        POP     BC
        EXX
        CALL    MOVE1
        POP     BC
        EXX
        JR      MOVE3
;
;==================================================
;  ROUTINES DE DEPLACEMENTS
;==================================================
;
```

*Sur la disquette d'accompagnement de ce numéro, vous trouverez un fichier TEXT DOS 3.3 du source de RAMdisk. Le fichier RAMDISK.COM a été obtenu par transfert de la disquette originale CP/M à l'aide du programme commercial Universal File Conversion. Vous utiliserez ce programme ou APDOS pour le transférer sur votre disquette CP/M.*

```
MOVE1:  LD      HL,(BANK)
        LD      (HL),A
        LD      HL,RDMAIN
        LD      DE,WRAUX
        JP      COPI1
;
MOVE2:  LD      HL,(BANK)
        LD      (HL),A
        LD      HL,RDMAIN
        LD      DE,WRMAIN
        JP      COPI1
;
MOVE3:  LD      (BK2),A
        LD      DE,WRAUX
        LD      BC,BKMAIN
        JR      COPI2
;
MOVE4:  LD      (BK2),A
        LD      DE,WRMAIN
        LD      BC,BKMAIN
        JR      COPI2
;
MOVE5:  LD      HL,(BANK)
        LD      (HL),A
        LD      DE,WRAUX
        LD      BC,BKAUX
        JR      COPI2
;
;==================================================
;ROUTINE DE COMMUTATION DE PAGES ET COPIE DE BLOC
;       RESIDE EN RAM AUX48
;==================================================
COPI2:  LD      (DE),A
        LD      (BC),A
        EXX
        LDIR
        LD      (BKAUX),A
        LD      (WRAUX),A
        RET
;
LG2     EQU     $-BLF   ;LONGUEUR DU CODE DE TRANSFERT
        .DEPHASE
LPROG   EQU     $-DEB   ;LONGUEUR DU PROGRAMME
;
;
;--------------------------------------------------
;
;ROUTINE DE DETOURNEMENT DES APPELS BIOS
;
DET0    EQU     $
DETF    EQU     DETOUR  ;ADRESSE D'IMPLANTATION
        .PHASE  DETF
;
READ:   LD      A,(0FEEFH)
        CP      RAMDSK
        JP      NZ,0FEC6H
        JP      RD
WRITE:  LD      A,(0FEEFH)
        CP      RAMDSK
        JP      NZ,0FECCH
        JP      WR
SELDSK: LD      A,RAMDSK
        CP      C
        JP      NZ,0FE97H
        LD      (0FEEFH),A
        LD      HL,0FA53H
        RET
;
LGDET   EQU     $-DETF
;
;--------------------------------------------------
;
;BLOC DE PARAMETRES DISQUES
;
        .DEPHASE
BLPAR:  DW      20H
        DFB     3,7,0
        DW      3BH             ;NB DE BLOCS-1 =59
        DW      2FH,0C0H,0CH,3
LGBL    EQU     $-BLPAR
;
        END
```

## Dump hexadécimal de 'RAMDISK.COM'

```
0100   00 00 00 21 00 F1 11 02 02 01 00 01 32 09 E0 1A
0110   ED A1 20 06 E2 1A 01 13 18 F5 32 08 E0 20 52 0E
0120   09 11 3C 01 CD 05 00 0E 01 CD 05 00 21 38 01 01
0130   04 00 ED B1 28 3B 18 6F 4F 59 6F 79 0D 0A 55 6E
0140   20 52 61 6D 64 69 73 6B 20 65 78 69 73 74 65 2E
0150   2E 2E 2E 0D 0A 66 61 75 74 2D 69 6C 20 6C 27 61
0160   6E 6E 75 6C 65 72 20 20 28 4F 2F 4E 29 3F 20 20
0170   24 21 03 01 11 03 01 01 C7 02 32 05 E0 ED B0 32
0180   03 E0 32 09 E0 21 02 02 11 00 F1 01 C8 01 ED B0
0190   21 00 30 3E E5 01 FF 0F 77 11 01 30 ED B0 32 08
01A0   E0 32 04 E0 32 02 E0 21 02 02 11 00 F1 01 0E 00
01B0   ED B0 21 CA 03 11 0E F1 01 23 00 ED B0 21 ED 03
01C0   11 63 FA 01 0F 00 ED B0 21 63 FA 22 5D FA 21 0E
01D0   F1 22 28 FA 21 19 F1 22 2B FA 21 24 F1 22 1C FA
01E0   0E 09 11 E8 01 C3 05 00 0D 0A 0D 0A 4D 3A 20 20
01F0   65 6E 20 73 65 72 76 69 63 65 2E 2E 2E 2E 2E 0D
0200   0A 24 AF 18 08 3E 02 18 04 32 08 E0 C9 32 09 E0
0210   32 82 F0 21 00 F1 ED 73 80 F0 F9 2A FA FE 22 83
0220   F0 2A EA FE 22 89 F0 32 03 E0 32 05 E0 3A 89 F0
0230   FE 12 30 2F FE 0F 38 02 D6 0F 21 83 E0 FE 0E 20
0240   08 21 8B E0 3E 0B 32 89 F0 22 87 F0 07 07 07 07
0250   57 2A 89 F0 AF 5F CB 1C 1F 6F 19 22 85 F0 CD 81
0260   F1 AF 21 3E 40 32 83 E0 2A 80 F0 F9 32 02 E0 32
0270   04 E0 32 EA F3 18 92 77 12 D9 ED B0 32 03 E0 32
0280   05 E0 C9 7C FE B0 3A 82 F0 CE 00 07 4F ED 5B 83
0290   F0 21 FF AF 3E F0 BA 30 02 16 01 AF ED 52 38 1B
02A0   23 0C CB 11 AF BC 20 16 3E 7F 95 38 11 0C 3C 54
02B0   5F 3A 82 F0 B7 28 07 D5 E5 18 03 3F CB 11 AF 47
02C0   CB 11 21 C9 F1 09 5E 23 56 EB E9 E9 F1 E9 F1 04
02D0   F2 E9 F1 11 F2 11 F2 11 F2 11 F2 1F F2 1F F2 39
02E0   F2 46 F2 63 F2 63 F2 39 F2 70 F2 2A 85 F0 11 00
02F0   F3 01 80 00 D9 CD B0 F2 21 00 F3 ED 5B 83 F0 01
0300   80 00 D9 C3 A5 F2 2A 85 F0 ED 5B 83 F0 01 80 00
0310   D9 18 7C 2A 85 F0 ED 5B 83 F0 01 80 00 D9 C3 A5
0320   F2 2A 83 F0 11 00 F3 01 80 00 D9 CD 9A F2 21 00
0330   F3 ED 5B 85 F0 01 80 00 D9 18 77 2A 83 F0 ED 5B
0340   85 F0 01 80 00 D9 18 3A 2A 83 F0 11 00 F3 C1 D9
0350   CD 80 F2 C1 D9 CD 9A F2 21 00 F3 ED 5B 85 F0 01
0360   80 00 D9 18 4D 2A 83 F0 ED 5B 85 F0 01 80 00 D9
0370   18 2A 2A 83 F0 ED 5B 85 F0 C1 D9 CD 80 F2 C1 D9
0380   18 1A 2A 87 F0 77 21 02 E0 11 05 E0 C3 75 F1 2A
0390   87 F0 77 21 02 E0 11 04 E0 C3 75 F1 32 83 E0 11
03A0   05 E0 01 08 E0 18 17 32 83 E0 11 04 E0 01 08 E0
03B0   18 0C 2A 87 F0 77 11 05 E0 01 09 E0 18 00 12 02
03C0   D9 ED B0 32 09 E0 32 05 E0 C9 3A EF FE FE 0C C2
03D0   C6 FE C3 00 F1 3A EF FE FE 0C C2 CC FE C3 03 F1
03E0   3E 0C B9 C2 97 FE 32 EF FE 21 53 FA C9 20 00 03
03F0   07 00 3B 00 2F 00 C0 00 0C 00 03 00 00 00 00 00
```

# TRES HAUTE RESOLUTION SUR DMP/IMAGEWRITER

Jean-Paul Payen

Vous êtes l'heureux possesseur d'une imprimante matricielle parallèle DMP ou de l'imprimante série ImageWriter de votre marque préférée. Alors vous allez pouvoir tracer vos courbes avec une résolution bien supérieure aux 280x192 points de l'écran que vous avez l'habitude de recopier avec votre programme de *hardcopy* habituel.
En effet, c'est 1280 points par ligne horizontale, et autant de points que vous voudrez en verticale (la seule limite étant la taille de la feuille de papier et aussi celle de la mémoire de votre Apple II) que vous pourrez obtenir avec ce programme TABLE.1280, écrit en assembleur pour avoir un temps d'éxécution convenable.

## Mode d'emploi

Il suffit de faire :

*RUN DEMO.TABLE.1280.*

Ce petit programme de démonstration charge TABLE.1280 et vous demande la hauteur de papier que vous voulez utiliser, ainsi que la densité du point.
La hauteur est *a priori* quelconque, mais on aura éventuellement des problèmes de taille mémoire si on demande trop. 15 à 20 cm assurent une bonne présentation pour faire des essais sur une feuille 21 x 29,7.
La densité correspond au nombre total de points que vous voudrez tracer. Plus elle est grande, plus la courbe est belle, mais plus aussi le temps d'exécution est long. Il n'y a pas de valeur idéale, cela dépend de la forme de la courbe. Faire un premier essai avec '1' ou '2' permet d'avoir une idée de la courbe obtenue sur l'écran haute-résolution. Au-delà les variables prendront la place mémoire de la page graphique et la courbe ne sera visible que sur

## Source 'T.TABLE.1280'
### Assembleur Big Mac

```
1   ********************************
2   *                              *
3   *   TABLE TRACANTE SUR D.M.P.  *
4   *                              *
5   *      ET IMAGEWRITER          *
6   *                              *
7   *                              *
8   *                              *
9   *           TABLE.2            *
10  *                              *
11  *          (Big Mac)           *
12  *                              *
13  *                              *
14  * J.P.PAYEN        le 26/02/86 *
15  *                              *
16  ********************************
17
18
19
20
21
22  DMP      =    $06        ;$80 si DMP, $00 si IMAGEWRITER
23  *  POKE 6,128  ou  POKE 6,0
24
25  INDEX    =    $07        ; Index ds le TABLEAU (0 à FF)
26  NBSAUT   =    $08        ; nbre de retour chariot
27  *  POKE 8,NSAUT%
28
29  NBK      =    $09        ; nbre total de pages)
30  *  POKE 9,NBK%
31
32  NBKCOUR  =    $1E        ; Val. courante de NBK
33
34  YCOURANT =    $18        ; Val.Y pour comparaisons (18.19)
35
36  SAUTCT   =    $1A        ; cteur de Sauts
37  MASK     =    $1B
38  TABADR   =    $FD        ; Adr. page courante TABLEAU
39  T0ADR    =    $E0        ; Adr. page 0 de TABLEAU
40  * $E2,E3,E4,E5 contiennent pages 1,2,3,4
41  * (partie hte de l'adressse)
42
43  ADRX     =    $EB        ; Adresse de X%(K)
44  ADRY     =    $ED        ; Adresse de Y%(K)
45  KX       =    $F9        ; Adr. courante de X%
46  KY       =    $FB        ;Adr. courante de Y%
47
48  TABLEAU0 =    $9000      ; Octets graph. à lire
49
50
51  CHRGET   =    $B1
52  PTRGET   =    $DFE3
53
```

*Ce source, sur la disquette d'accompagnement, est sauvegardé en format TEXT pour permettre l'édition avec un autre assembleur*

l'imprimante, ce qui d'ailleurs est le but de l'opération. Avec une densité de '4' on remplit complétement la mémoire de l'Apple II. (Il faudra alors supprimer le LOMEM de la ligne 25 qui protège la page graphique).

Quand le tracé est fini taper *<RETURN>* puis suivant votre imprimante tapez I (magewriter) ou D (MP) quand le programme le demande : il faudra auparavant avoir alimenté en papier l'imprimante, en tenant compte du fait qu'elle le déroule à l'envers. Ceci pour avoir la courbe dans un sens tel que l'on pourra ajouter du texte par-dessus si on le désire.

Le programme de démonstration va calculer les points représentatifs de la fonction Y (X) , à l'intérieur d'une fenêtre décrite par les variables XN,XM,YN,YM (lignes 100-160). Il est de la responsabilité du programmeur de modifier cette partie du programme qui a été volontairement choisie très simple. Il est d'ailleurs possible de choisir une fenêtre et une hauteur d'impression telles que l'on puisse utiliser du papier millimétré avec une bonne correspondance entre l'échelle de la courbe et celle du papier. L'auteur en a fait l'expérience.

Ligne 300-480, le programme calcule le nombre de retours chariots de l'imprimante (NSAUT%), les échelles compte tenu de la fenêtre, le nombre total de points (KM) et le nombre de pages mémoire correspondantes (NBK%). KM est un multiple de 256 pour simplifier l'analyse du programme machine.

Ligne 500-590 : calcul des points des axes horizontaux et verticaux. Cette partie peut etre développée si l'on veut faire figurer les petits traits d'unité sur les axes. On veillera alors à utiliser un nombre de points multiple de 256 et d'en tenir compte dans le calcul de NBK%.

Ligne 600-670 : calcul des points de la courbe à tracer. Chaque point est représenté par un couple

```
54  SLOT     =    1            ;Pour la DMP uniquement
55  DEVICE   =    $C080+16*SLOT ;Pour la DMP (carte PIC)
56  PIC      =    $C0C1+256*SLOT ;Idem
57  COUT     =    $FDED        ;Sortie standard (IMAGEWRITER)
58
59
60
61
62            ORG  $8E30        ; CALL 36400
63
64
65            JSR  CHRGET       ; Récupère adresse
66            JSR  PTRGET       ; de X%(0)
67            STA  ADRX
68            STY  ADRX+1
69
70            JSR  CHRGET
71            JSR  PTRGET       ; Y%(0)
72            STA  ADRY
73            STY  ADRY+1
74
75
76            LDX  #>TABLEAU0 ; $90 (à modifier
77  *   si autre implantation de TABLEAU0)
78            STX  T0ADR+1
79            INX
80            STX  T0ADR+2      ; $91
81            INX
82            STX  T0ADR+3      ; $92
83            INX
84            STX  T0ADR+4      ; $93
85            INX
86            STX  T0ADR+5      ; $94
87            LDA  #<TABLEAU0 ; $00
88            STA  T0ADR        ; $9000 ds E0.E1
89            STA  TABADR
90
91            LDY  #0
92            STY  SAUTCT       ; init. cpteur de sauts à 0
93            STY  YCOURANT
94            STY  YCOURANT+1 ; mise à 0 de YCOURANT
95
96
97
98 * Envoi des caractères de controle à l'imprimante *
99
100
101           LDA  #$1B
102           JSR  OUT
103           LDA  #"P"
104           JSR  OUT          ; -->  (ESC P)
105
106 * Mise en caractères proportionnels 1280 pts par ligne
107
108           LDA  #$1B
109           JSR  OUT
110           LDA  #"r"
111           JSR  OUT          ; --> 1B 72 (ESC r)
112
113 * Saut de ligne arrière
114
115           LDA  #$00
116           JSR  OUT          ; -> 00
117           LDA  #$0D
118           JSR  OUT          ; -> position à la marge gauche
```

```
119
120            LDA  #$1B
121            JSR  OUT
122            LDA  #"T"
123            JSR  OUT
124            LDA  #$31
125            JSR  OUT
126            LDA  #$36
127            JSR  OUT          ; -> 1B 54 31 36
128
129  * Esc T 49 54 = Esc T 48+1 48+6
130  *  -> 16/144 = 1/9 pouce pour L.F.
131
132
133            LDA  #$1B
134            JSR  OUT
135            LDA  #$3E
136            JSR  OUT          ; -> 1B 3E
137
138  * Esc > : unidirectionnelle
139
140 ************************************************
141
142
143
144
145 * Mise à 0 de $9000 --> $94FF
146
147  MISEZERO LDA  #0
148           LDX  T0ADR+1
149           STX  TABADR+1
150           LDY  #0
151           LDX  #5
152  ENCORE   STA  (TABADR),Y
153           INY            ; Y=0 ?
154           BNE  ENCORE    ; non
155           INC  TABADR+1
156           DEX
157           BNE  ENCORE
158
159           LDA  T0ADR+1
160           STA  TABADR+1  ; réinitialise TABADR
161
162
163           LDA  #%10000000
164           STA  MASK
165           BNE  CH1
166
167
168  RELMONT1 BNE  MISEZERO
169
170
171  FORBIT07 LSR  MASK
172           BEQ  ESCG1280
173  CH1      LDX  NBK
174           STX  NBKCOUR   ; initial. multiple de 256 pts
175           LDX  #0        ; 256 pts à compter
176
177           LDA  ADRX      ; Réinitial. KX et KY
178           STA  KX
179           LDA  ADRX+1
180           STA  KX+1
181           LDA  ADRY
182           STA  KY
183           LDA  ADRY+1
```

X%(K),Y%(K). Il est tout à fait possible de tracer ensemble plusieurs fonctions, la seule limite étant toujours la taille de la mémoire pour stocker tous ces points. Le travail essentiel de TABLE.1280 consistera donc à traduire la position de ces points en octets à envoyer à l'imprimante.

Le programme de démonstration passe ensuite par une visualisation de la courbe. Cette partie étant facultative.
Il se termine par l'appel de TABLE.1280 : il passe NSAUT% et NBK% comme paramètres, ainsi que l'adresse de début des tableaux X%(K) et Y%(K).

## Fonctionnement de TABLE.1280

Écrit avec l'assembleur Big Mac, ce programme commence par récupérer l'adresse des tableaux X%(K) et Y%(K). Il initialise ensuite un certain nombre de pointeurs, notamment l'adresse de l'espace mémoire appelé TABLEAU0 qui sert à stocker temporairement 1280 octets à envoyer à chaque passe de la tête de l'imprimante. Cette partie est la seule se référant à une adresse absolue, le reste du programme étant totalement relogeable (pas de JSR ni de JMP). Donc, si l'on veut changer la place en mémoire de TABLE.1280 et du TABLEAU0 sans réassembler le source, il suffira de changer les valeurs $90 et $00 utilisées au début, pour les remplacer par l'adresse de son choix.
Le programme envoie ensuite les codes de contrôle à l'imprimante : (ESC-P) pour la mettre en mode proportionnel (1280 points par ligne), (ESC-r) pour le défilement arrière du papier, (ESC-T 49 54) règle l'interligne à 1/9 pouce, et enfin (ESC->) pour une impression unidirectionnelle, sinon les axes verticaux seraient mal tracés. Il devrait être possible en modifiant ces caractères de contrôle d'utiliser d'autres imprimantes.
Suit la partie de recherche et

d'impression qui sera expliquée plus en détail et le programme se termine en remettant l'imprimante dans son état habituel.

## Recherche et impression

Le K$^{ième}$ point, calculé par le programme Basic est donc défini par ses coordonnées X%(K), Y%(K) dans le plan de la feuille de papier.

Prenons par exemple les 8 premières lignes en bas de la feuille. YCOURANT est mis à 0 pour commencer. Le programme va chercher s'il existe une ou plusieurs valeurs Y%(K) égales à YCOURANT. Quand il en trouve une, il va lire X%(K) et va ranger 1000 0000 à l'adresse TABLEAU0 + X%.

Il passe alors à YCOURANT = 1 ; recommence le même travail et rangera alors 0100 0000. Etc. jusqu'à la 8$^{ème}$ ligne. S'il se trouve 2 valeurs différentes de Y% pour le même X%, par exemple Y = 5 et Y = 3 pour X = 149, en faisant un ORA, on trouvera à l'adresse TABLEAU0+149, 0001 0100.

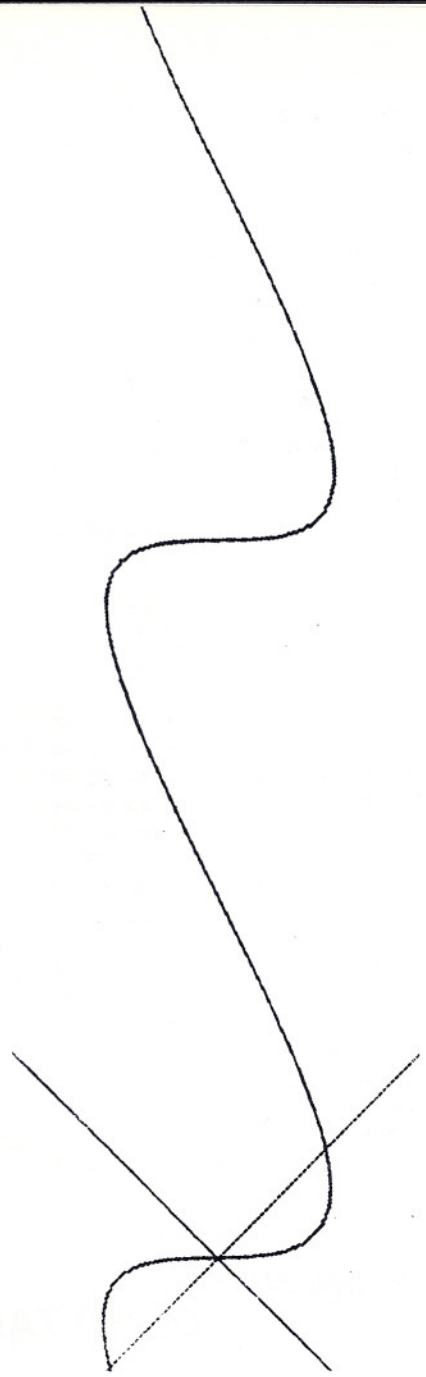
Le programme prévient alors l'imprimante qu'il va lui envoyer 1280 positions de points (ESC G 49 50 56 48), et elle imprime les 8 * 1280 points correspondant aux 1280 octets ainsi définis. On regarde si la recherche est terminée (comparaison avec NSAUT%) et on recommence le travail si ce n'est pas le cas, en remettant au préalable à 0 tout l'espace mémoire réservé pour TABLEAU0.

Ce programme fonctionne avec l'interface parallèle PIC d'Apple ou avec une interface série branchée sur ImageWriter, les codes de contrôle des deux imprimantes étant les mêmes.

```
184            STA  KY+1
185
186  CHERCHE   LDY  #0
187            LDA  (KY),Y      ; lit 1er octet de Y%(K)
188            CMP  YCOURANT
189            BNE  KSUIVANT    ; s'il est différent on saute
190            INY
191            LDA  (KY),Y      ; lit 2e octet de Y%(K)
192            CMP  YCOURANT+1
193            BNE  KSUIVANT    ; s'il est différent on saute
194
195            LDY  #0          ; il est idem
196            LDA  (KX),Y      ; lit 1er octet de X%(K) -> n[page
197            TAY
198            LDA  T0ADR+1,Y   ; charge n[ page
199            STA  TABADR+1    ; mis ds adr. courante
200            LDY  #1
201            LDA  (KX),Y      ; lit 2e octet de X%(K)
202            TAY
203            LDA  (TABADR),Y
204            ORA  MASK
205            STA  (TABADR),Y ; met le bit ds TABLEAU
206
207  KSUIVANT  INC  KX
208            BNE  I1
209            INC  KX+1
210  I1        INC  KX
211            BNE  I2
212            INC  KX+1
213
214  I2        INC  KY
215            BNE  I3
216            INC  KY+1
217  I3        INC  KY
218            BNE  I4
219            INC  KY+1

221  I4        DEX              ; dernier K ?
222            BNE  CHERCHE     ; non
223            DEC  NBKCOUR     ; dernier NBK ?
224            BNE  CHERCHE     ; non
225
226            INC  YCOURANT+1 ; ligne suivante
227            BNE  FORBIT07
228            INC  YCOURANT
229            BNE  FORBIT07
230
231
232
233  RELMONT2  BNE  RELMONT1
234
235
236  ESCG1280  LDA  #$1B
237            JSR  OUT
238            LDA  #"G"
239            JSR  OUT
240            LDA  #$31
241            JSR  OUT
242            LDA  #$32
243            JSR  OUT
244            LDA  #$38
245            JSR  OUT
246            LDA  #$30
247            JSR  OUT
248
```

```
249  * 1B 47 31 32 38 30
250
251  * Esc G  48+1 48+2 48+8 48+0
252  *        1280 : 1280 positions de points
253
254
255
256           LDA  #>TABLEAU0
257           STA  TABADR+1    ; réinit. TABADR
258           LDA  #<TABLEAU0
259           STA  TABADR
260
261  *  Lecture des 1280 octets de TABLEAU0
262
263
264
265           LDA  #0
266  STAINDEX       ; Initialise INDEX
267
268  IMPRIME  LDY  INDEX
269           LDA  (TABADR),Y ; lit TABLEAU
270           JSR  OUT        ; 8 pts imprimés
271           INC  INDEX      ; INDEX <= $FF
272           BNE  IMPRIME    ; oui, au suivant
273
274           INC  TABADR+1   ; non à la page suivante
275           LDA  TABADR+1
276           CMP  #>TABLEAU0+$500 ; dernier ?
277           BNE  IMPRIME    ; non
278
279
280  *  Impression des 1280 pts terminée
281
282
283           LDA  #$0D
284           JSR  OUT
285           BIT  DMP
286           BPL  NOLF       ;Pas LF si IMAGEWRITER
287           LDA  #$0A
288           JSR  OUT
289  NOLF     EQU  *
290
291  *  CTRL-M : saut de ligne
292
293
294
295           INC  SAUTCT
296           LDA  SAUTCT
297           CMP  NBSAUT     ; dernière ligne ?
298           BEQ  FIN        ; oui
299
300
301
302           BNE  RELMONT2   ; non --> Mise à zéro
303
304  * FIN remise en état normal de l'imprimante
305
306  FIN      LDY  #0
307           LDA  #$1B
308           JSR  OUT
309           LDA  #$41
310           JSR  OUT        ; -> 1B 41
311
312  * Esc A : interligne de 1/6
313
```



## Récapitulation 'TABLE.1280'

Après avoir saisi ce code sous moniteur, vous le sauvegarderez par
BSAVE TABLE.1280,A$8E30,L$18F

```
8E30- 20 B1 00 20 E3 DF 85 EB
8E38- 84 EC 20 B1 00 20 E3 DF
8E40- 85 ED 84 EE A2 90 86 E1
8E48- E8 86 E2 E8 86 E3 E8 86
8E50- E4 E8 86 E5 A9 00 85 E0
8E58- 85 FD A0 00 84 1A 84 18
8E60- 84 19 A9 1B 20 AF 8F A9
8E68- D0 20 AF 8F A9 1B 20 AF
8E70- 8F A9 F2 20 AF 8F A9 00
8E78- 20 AF 8F A9 0D 20 AF 8F
```

```
8E80- A9 1B 20 AF 8F A9 D4 20
8E88- AF 8F A9 31 20 AF 8F A9
8E90- 36 20 AF 8F A9 1B 20 AF
8E98- 8F A9 3E 20 AF 8F A9 00
8EA0- A6 E1 86 FE A0 00 A2 05
8EA8- 91 FD C8 D0 FB E6 FE CA
8EB0- D0 F6 A5 E1 85 FE A9 80
8EB8- 85 1B D0 06 D0 E0 46 1B
8EC0- F0 63 A6 09 86 1E A2 00
8EC8- A5 EB 85 F9 A5 EC 85 FA
8ED0- A5 ED 85 FB A5 EE 85 FC
8ED8- A0 00 B1 FB C5 18 D0 1C
8EE0- C8 B1 FB C5 19 D0 15 A0
8EE8- 00 B1 F9 A8 B9 E1 00 85
8EF0- FE A0 01 B1 F9 A8 B1 FD
8EF8- 05 1B 91 FD E6 F9 D0 02
8F00- E6 FA E6 F9 D0 02 E6 FA
8F08- E6 FB D0 02 E6 FC E6 FB
8F10- D0 02 E6 FC CA D0 C1 C6
8F18- 1E D0 BD E6 19 D0 9F E6
8F20- 18 D0 9B D0 97 A9 1B 20
8F28- AF 8F A9 C7 20 AF 8F A9
8F30- 31 20 AF 8F A9 32 20 AF
8F38- 8F A9 38 20 AF 8F A9 30
8F40- 20 AF 8F A9 90 85 FE A9
8F48- 00 85 FD A9 00 85 07 A4
8F50- 07 B1 FD 20 AF 8F E6 07
8F58- D0 F5 E6 FE A5 FE C9 95
8F60- D0 ED A9 0D 20 AF 8F 24
8F68- 06 10 05 A9 0A 20 AF 8F
8F70- E6 1A A5 1A C5 08 F0 02
8F78- D0 A9 A0 00 A9 1B 20 AF
8F80- 8F A9 41 20 AF 8F A9 0D
8F88- 20 AF 8F A9 1B 20 AF 8F
8F90- A9 3C 20 AF 8F A9 0A 20
8F98- AF 8F A9 1B 20 AF 8F A9
8FA0- 66 20 AF 8F A9 1B 20 AF
8FA8- 8F A9 4E 20 AF 8F 60 24
8FB0- 06 30 03 4C ED FD 8D 90
8FB8- C0 AD C1 C1 30 FB 60
```

```
314          LDA  #$0D
315          JSR  OUT        ; -> 0D marge gauche
316          LDA  #$1B
317          JSR  OUT
318          LDA  #$3C
319          JSR  OUT        ; -> 1B 3C
320
321  * Esc < : impression bidirectionnelle
322
323          LDA  #$0A
324          JSR  OUT        ; -> 0A : saut de ligne LF
325          LDA  #$1B
326          JSR  OUT
327          LDA  #$66
328          JSR  OUT        ; --> 1B 66 (ESC f)
329
330  * Saut de ligne avant
331
332          LDA  #$1B
333          JSR  OUT
334          LDA  #$4E
335          JSR  OUT        ; --> 1B 4E (ESC N)
336
337  *  Caractère PICA
338
339
340          RTS             ;Fin
341
342  *  Envoie vers l'imprimante
343  *  ------------------------
344
345  OUT     BIT  DMP        ;Identifie l'imprimante
346          BMI  DMPOUT
347          JMP  COUT       ;IMAGEWRITER
348  DMPOUT  STA  DEVICE     ;DMP
349  ATTENTE LDA  PIC
350          BMI  ATTENTE
351          RTS
```

## Programme 'DEMO.TABLE.1280'

```
10  REM         DEMO.TABLE.2
15  REM
16  REM  -----------------------
20  HIMEM: 36400
25  LOMEM: 16385
30 D$ =  CHR$ (4)
35  GOSUB 2000: REM EXPLICATIONS
40  PRINT D$"BLOAD TABLE.1280"
50  REM
60  REM
70  REM  -----------------------
80  REM  FONCTION & FENETRE A
90  REM DEFINIR PAR L'UTILISATEUR
100  REM -----------------------
110  REM
120  DEF  FN Y(X) = 1 -  EXP ( - .2 * X) *
     COS (X)
125 PI = 3.14159
130 XN = 0: REM      X mini
140 XM = 5 * PI: REM X maxi
150 YN = 0: REM      Y mini
160 YM = 1.6: REM    Y maxi
170  REM
180  REM
190  REM  --------------
200  REM  CHOIX INITIAUX
205  REM  --------------
210  HOME
215  INPUT "HAUTEUR EN CM (DE 10 A 30) ";H
220  PRINT
235  PRINT
240  INPUT "DENSITE DU TRACE  ( 1 A 2) ";F%
250  PRINT :AX = 0
260  INPUT "TRACE DES AXES (O/N) ? ";A$: IF
     A$ = "O" THEN AX = 1
270  PRINT : HTAB 6: INVERSE : PRINT "ENVIRO
     N ";3 * F%;" MINUTES D'ATTENTE": NORMAL

280  PRINT "ATTENDRE LES 2 BIP POUR PRESSER
     'RETURN'"
285  REM
290  REM  --------------------
300  REM  CALCUL DE LA PLACE A RESERVER DANS
          LA MEMOIRE
310  REM --------------------
320  REM
330  IF (XN > 0) OR (XM < 0) OR (AX = 0) THE
     N PV% = 0: GOTO 360: REM PAS D'AXE VERT
```

```
ICAL A TRACER
340 PV% = 1 + H / 8.8: REM  NBRE PAGES AXE V
ERTICAL
350  REM
360 NSAUT% = H / 0.275: REM  NBRE SAUTS TETE
 IMPRIM.
370 EY = 8 * NSAUT% / (YM - YN): REM   ECHEL
LE EN Y
380  REM
390  IF (YN > 0) OR (YM < 0) OR (AX = 0) THE
N PH% = 0: GOTO 420: REM PAS D'AXE HORI
ZONTAL A TRACER
400 PH% = F%: REM  NBRE PAGES POUR AXE HORIZ
ONTAL
410  REM
420 EX = 1280 / (XM - XN): REM ECHELLE EN X
430 NBK% = PV% + PH% + 5 * F%
440  REM  NBRE DE PAGES MEM A RESERVER
450 KM = 256 * NBK%: REM  NBRE TOTAL DE POIN
TS
480  DIM X%(KM),Y%(KM)
490  REM
495  REM CALCUL DE L'AXE VERTICAL
500  REM
505  IF PV% = 0 THEN JY =  - 1: GOTO 555
510 X0% =  - XN * EX + 0.5
515  FOR JY = 1 TO 256 * PV%
520 X%(JY - 1) = X0%
525 Y%(JY - 1) = JY
530  NEXT JY
535 JY = JY - 2
540  REM
545  REM CALCUL AXE HORIZONTAL
550  REM
555  IF PH% = 0 THEN KN = JY + 1: GOTO 610
560 Y0% =  - YN * EY + 0.5
565  FOR JX = 1 TO 256 * F%
570 X%(JX + JY) = 5 * JX / F%
575 Y%(JX + JY) = Y0%
580  NEXT JX
585 KN = JX + JY
590  REM
595  REM
600  REM CALCUL DE LA COURBE
605  REM
610 S = (XM - XN) / 1280 / F%: REM  PAS DE C
ALCUL
615 X = XN - S
620  FOR K = KN TO KM - 1
630 X = X + S
640 Y =  FN Y(X)
645  IF Y < YN OR Y > YM THEN 670: REM  PTS
HORS FENETRE
650 X%(K) = (X - XN) * EX + 0.5
660 Y%(K) = (Y - YN) * EY + 0.5
670  NEXT K: REM
745  REM  -----------
750  REM   TRACE HGR
755  REM  -----------
760  HGR
765  POKE 49234,0
770  HCOLOR= 3
780  FOR K = 0 TO KM - 1
790 X = X%(K) / 4.6
800 Y = 191 - Y%(K) / NSAUT% * 23.8
810  IF Y < 0 THEN 840
830  HPLOT X,Y
840  NEXT K: REM
850  PRINT  CHR$ (7): PRINT  CHR$ (7)
860  INPUT A$
870  REM
900  TEXT : HOME
910  PRINT "CALCUL TERMINE": PRINT : PRINT
920  PRINT "POUR COMMENCER A IMPRIMER"
930  PRINT : PRINT "(I) POUR L'IMAGEWRITER":
 PRINT : PRINT "(D) POUR LA DMP"
940  PRINT : INVERSE : PRINT "ATTENTION, LE
PAPIER DEFILE A L'ENVERS": NORMAL
950  GET A$: IF A$ <  > "I" AND A$ <  > "i"
AND A$ <  > "D" AND A$ <  > "d" GOTO 95
0
960  IF A$ = "D" OR A$ = "d" THEN  POKE 6,12
8
970  IF A$ = "I" OR A$ = "i" THEN  POKE 6,0
980  PRINT :A =  PEEK (49232)
990  REM
1000  REM      IMPRESSION
1005  REM ---------------
1010  POKE 8,NSAUT%
1020  POKE 9,NBK%
1025  IF A$ = "D" OR A$ = "d" GOTO 1040
1030  PRINT D$"PR#1": PRINT  CHR$ (9)"Z"
1035  REM  Ajouter: PRINT  CHR$ (27); CHR$ (
68); CHR$ (128); CHR$ (0) si vous n'av
ez aucun saut de ligne.
1040  CALL 36400,X%(0),Y%(0)
1045  PRINT
1050  PRINT D$"PR#0"
1060  TEXT
1070  HOME
1080  PRINT "POUR OBTENIR LA MEME FONCTION"
1085  PRINT
1090  PRINT "TAPER ";: INVERSE : PRINT " RUN
";: NORMAL
1095  PRINT
1100  PRINT : PRINT : PRINT "POUR MODIFIER L
A FENETRE"
1101  PRINT "OU LA FONCTION:"
1102  PRINT
1103  PRINT " CHANGER LES LIGNES"
1104  PRINT "120 A 160 SELON VOS DESIRS": PR
INT
1110  END
1990  REM
1995  REM  ------------
2000  REM  EXPLICATIONS
2005  REM  ------------
2010  HOME
2020  PRINT "CECI EST UN PRGM DE DEMONSTRATI
ON"
2030  PRINT : PRINT "ON POURRA OBTENIR DAVAN
TAGE DE POINTS"
2040  PRINT "EN SUPPRIMANT LA REPRESENTATION
"
2050  PRINT "GRAPHIQUE SUR L'ECRAN"
2060  PRINT
2070  PRINT "IL FAUDRA ALORS SUPPRIMER AUSSI
"
2080  PRINT "LE 'LOMEM' DE LA LIGNE 25"
2090  RETURN
```

## ou ne plus perdre d'anciennes versions...

Voici un petit programme destiné à faciliter l'écriture de programmes Basic. En fait, il s'agit de deux fichiers TEXT écrits avec AppleWriter.

Avant d'écrire un programme Applesoft, il faut faire

*RUN START L/S*

ou taper au clavier les 2 POKEs qui transformeront les commandes LOAD et SAVE en #OAD et #AVE. Ceci afin d'éviter des erreurs.
Ensuite, pour charger ou sauvegarder un programme, il suffit :

- pour charger un programme, de faire :

*EXEC LOAD*

On donne le nom du programme et il se charge en mémorisant son nom en $300... avec un numéro de version.

- pour sauvegarder, un :

*EXEC SAVE*

lira automatiquement le nom du programme en machine, ainsi que son numéro de version, qui sera incrémenté ; puis, la commande SAVE est exécutée. Si aucun nom n'est en mémoire, il faudra l'entrer au clavier.
Ceci devrait éviter les 'écrasements involontaires'.

NDLR : Nous avons remanié les deux fichiers EXEC pour les rendre un peu plus fiables. A noter que c'est un moyen gourmand en place sur la disquette, et qu'il faut de temps en temps nettoyer la disquette. Pourquoi ne pas faire un fichier EXEC qui efface toutes les versions dont le numéro est inférieur à un certain rang ?

## Programme 'START L/S'

```
10  REM START L/S
15  REM NEFONCTIONNE PAS SI LE DOS EST SUR LA CARTE LANGA
    GE
20  TEXT : HOME
30  POKE 43144,35: REM LOAD DEVIENT #OAD
40  POKE 43148,35: REM SAVE DEVIENT #AVE
50  PRINT " LOAD ET SAVE SONT INVALIDES"
60  PRINT " UTILISEZ EXEC LOAD ET EXEC SAVE"
```

DOS 3.3 seulement...

## Fichier EXEC 'LOAD'

*NB : Les • symbolisent des espaces. Il y a un retour chariot à la fin de ce fichier.*

```
TEXT:HOME:VTAB 4:POKE 768,0
POKE 118,0:POKE 43699,0:INPUT "•••••••••••••••••••••••••
    •••••••••••••••LOAD:•";NP$:A$=NP$:POKE 43699,1
ER=1:PRINT CHR$(4)"VERIFY"A$:ER=0:POKE 768,137
IF NOT ER THEN FOR I=1 TO LEN(A$):IF MID$(A$,I,1)<>"." TH
    EN NEXT
IF I=1 OR I>36 THEN ER=1
IF NOT ER THEN NV=VAL(MID$(A$,I+1)):IF NV<0 OR NV>99 THEN
     ER=1
IF NOT ER THEN A$=LEFT$(A$,I-1):PRINT "PROGRAMME: "A$" -
    VERSION: "NV:A$=A$+".":IF NV<10 THEN A$=A$+"0"
IF NOT ER THEN A$=A$+STR$(NV):LN=LEN(A$):POKE 810,LN:POKE
     811,NV:FOR I=1 TO LN:POKE 768+I,ASC(MID$(A$,I,1)):N
    EXT:PRINT CHR$(4)" #OAD ";NP$
IF ER THEN PRINT "ERREUR - CHARGEMENT IMPOSSIBLE"
```

## Fichier EXEC 'SAVE'

*NB : Les • symbolisent des espaces. Il y a un retour chariot à la fin de ce fichier.*

```
TEXT:HOME:VTAB 4:ER=0
MN=0:IF PEEK(768)<>137 THEN MN=1:POKE 118,0:POKE 43699,0:
    INPUT "•••••••••••••••••••••••••••••••••••••••SAVE:
    •";NP$:POKE 43699,1
IF MN THEN ER=1:LN=LEN(NP$):FOR I=1 TO LN:IF MID$(NP$,I,1
    )<>"." THEN NEXT:IF LN<=36 AND LN>0 THEN ER=0
IF MN AND NOT ER THEN NP$=NP$+".00":LN=LN+3:POKE 810,LN:P
    OKE 811,0:FOR I=1 TO LN:POKE 768+I,ASC(MID$(NP$,I,1)
    ):NEXT:POKE 768,137
IF NOT MN THEN PRINT "SAUVEGARDE AUTOMATIQUE":NP$="":LN=P
    EEK(810):FOR I=1 TO LN:NP$=NP$+CHR$(PEEK(768+I)):NEX
    T:NP$=LEFT$(NP$,LN-2):NV=PEEK(811)+1:IF NV>99 THEN E
    R=1
IF NOT MN AND NOT ER THEN POKE 811,NV:IF NV<=9 THEN NP$=N
    P$+"0"
IF NOT MN THEN NP$=NP$+STR$(NV)
IF ER THEN PRINT "ERREUR - SAUVEGARDE IMPOSSIBLE"
IF NOT ER THEN PRINT "PROGRAMME: "LEFT$(NP$,LEN(NP$)-3)"
    - VERSION: "NV:PRINT CHR$(4)" #AVE ";NP$
```

# Postscript

Calcul : 9690 millisecondes.

*Christian Piard*

Cette présentaton du langage PostScript est une forme de réponse aux lecteurs qui s'interrogaient sur le programme générant certains titres de Pom's. Les indications données ici seront directement utilisables par ceux qui ont la disposition d'une LaserWriter. En revanche, le Macintosh n'est pas indispensable, un Apple // (ou tout autre ordinateur doté d'une interface Série) et un traitement de texte conviendront, c'est d'ailleurs la configuration retenue par Pom's. L'heureux utilisateur de la LaserWriter lira avec intérêt "Macintosh Inside LaserWriter" la bible de PostScript en 450 pages.

## PostScript

Ce langage est intégré à la LaserWriter d'Apple et d'autres imprimantes à laser l'ont en option. Communément utilisé sur les machines de photocomposition (linotype par exemple), il tire sa souplesse et sa portabilité (chère aux Pascaliens...) du fait que les ordres ne sont pas réellement des commandes d'imprimantes mais plutôt une description de la page à imprimer. Le langage s'adapte aux caractéristiques de l'imprimante, adaptation que le programmeur peut contrôler (résolution horizontale et verticale, trame, couleur, saturation, dimensions, polices de caractères, niveaux de gris...). Il est bien entendu particulièrement orienté vers le graphisme mais trigonométrie, logarithmes, opérations logiques, traitements des chaînes ne lui font pas défaut : près de 240 opérateurs pour définir toute forme à imprimer.

Sur le Macintosh, les 'Drivers' transforment les textes, formes et motifs créés par MacWrite, MacDraw, etc. en des ordres PostScript, interprétés dans la LaserWriter qui les reçoit sous forme ASCII.

## Les piles

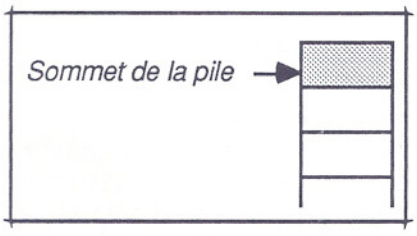
Possesseurs de calculatrices HP et pratiquants du Forth sont ici en pays de connaissance, PostScript est basé sur la gestion d'une pile ou plutôt de quatre piles :

- **pile dictionnaire** qui est la liste des commandes connues du langage, liste complétée par l'utilisateur en fonction de ses besoins ;
- **pile opérateur** qui peut contenir toutes les sortes d'objets PostScript sur lesquels on travaille. On y met arguments et opérandes et on y récupère les résultats. Ces objets peuvent être des entiers (jusqu'à ± 2 147 483 647 de la base 2 à la base 36 !) des réels de $10^{-38}$ à $10^{38}$, des tableaux à une seule dimension mais 65535 éléments comptés à partir de 0, des chaînes, des dictionnaires, etc. ;
- **pile d'exécution**, inaccessible à l'utilisateur, c'est la pile de travail de PostScript ;
- **pile d'état graphique**, également interne mais accessible à l'utilisateur qui peut préserver et restaurer des formes, polices, niveau de gris, épaisseur de trait en cours.

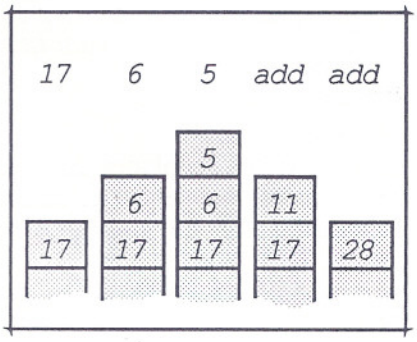
Il convient de préciser avant les deux exemples le fonctionnement de la pile et les possibilités de *définitions*.

## La pile

La pile peut être considérée comme une série de cellules superposées dont seule la plus haute (et donc la plus récemment empilée) est accessible.




Une addition telle que 17 + 6 + 5 se présente ainsi : *17 6 5 add add* la pile étant constituée à chaque étape comme suit :




*17 6 add 5 add* est équivalent, le résultat se trouvant toujours dans la cellule la plus haute. L'opération (3*2)^(4-1) se programmerait ainsi : *2 3 mul 4 1 sub exp* ou *4 1 sub 2 3 mul exch exp* ce qui est plus long mais permet de présenter *exch* opération d'échange les deux premières valeurs de la pile.

Rechercher le nombre d'éléments d'un tableau revient à faire *[56 78 34] length* et on recupère 3 au sommet de la pile. Même principe pour une chaîne dont on veut connaître le nombre de caractères : *(Pom's) length* laisse 5 (le nombre de caractères) dans la pile.

Le résultat peut occuper plus d'une cellule : dans la police de caractères courante, quelles seraient les hauteurs et largeur du mot Pom's ? *(Pom's) stringwidth* donne dans la cellule la plus haute la hauteur et dans la suivante la largeur.

## Définitions

PostScript permet de compléter le dictionnaire en associant des valeurs ou des procédures à des noms, possibilité exploitée dans les exemples.

*/nombre 45 def*
*/tabl [15 25 89] def*
*/unnom (Pom's) def*

Si, dans la suite du programme on invoque nombre, tabl ou un nom, on empilera le nombre 45, le tableau de 3 éléments, la chaîne de 5 caractères ou tout autre objet prédéfini. Ceci était la première façon de définir un nom du dictionnaire.

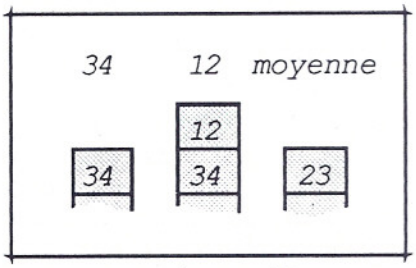
Voici la seconde :

*/op {dup} def*
*/moyenne {add 2 div} def*
*/puiss2 {dup mul} def*

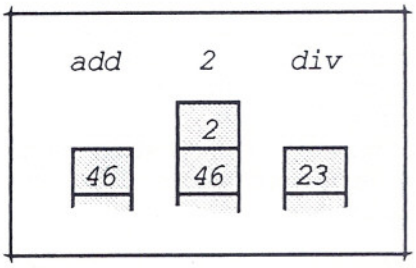
*op*, du fait des accolades dans la définition est maintenant associé à l'opération dup qui duplique le premier élément de la pile :

*56 78 12 op* donne dans la pile 56 78 12 12.

*34 12 moyenne* laissera 23 dans la pile :




moyenne qui se décompose ainsi :




idem pour *puiss2* qui élève au carré par duplication suivie d'une multiplication.

## Programme TITRE

Le premier exemple consiste en la réalisation du titre de cet article qui est irréalisable sans la commande directe de la LaserWriter, aucun programme pour Macintosh n'étant encore capable de générer ce genre de fantaisie.

Des caractères Times (la police utilisée pour le texte de cet article) de hauteur et d'inclinaison variable formeront le mot 'Postscript' centré sur une page. Deux partie dans ce petit programme : des définitions puis une boucle d'impression.

Le premier ordre *usertime* (ligne 1) consiste à empiler la valeur en cours du compteur de l'imprimante, compteur incrémenté mille fois chaque seconde, ceci pour connaitre le temps nécessaire au calcul.

*Police* (2) est associé à l'opération de recherche dans le dictionnaire de la police Times-Roman et *chaine* (3) au mot 'Postscript'. *Matrice* (4), un tableau de six nombres sera empilé pour passer à *makefont* les arguments nécessaires à la modification de la police standard Times-Roman.

Dans ce tableau, nous trouvons successivement exprimés en points :

- la largeur de base de la police de caractères (bien sûr chaque caractère n'a pas pour autant une chasse fixe), cette largeur pouvant être négative, ce qui conduit à écrire de droite à gauche ;
- un nombre qui affecte l'horizontalité du texte (tangente de l'angle entre ligne d'écriture et horizontale) ;
- tangente de l'angle des caractères avec la verticale ;
- hauteur de base de la police qui peut aussi être négative pour obtenir des lettres tête en bas ;
- puis les vecteurs x et y s'additionnant à la position théorique d'écriture.

*Carac* (5) est associé à la création d'une chaîne d'un caractère qui servira de tampon d'impression puisqu'on écrira lettre par lettre. *Numcarac* (6) est défini comme valant 0 et gardera le numéro de la lettre en cours d'impression. *Demilong* (7) qui est le nombre de caractères divisé par deux permettra de répartir la hauteur et l'inclinaison équitablement.

Premier travail, mesurer la chaîne pour pouvoir la centrer : on recherche la police voulue (8), on empile le tableau qui donne une largeur de 40 et une hauteur de départ de 75, *makefont* se charge de modifier la matrice de paramètres standard (en fait une copie...), et *setfont* enfin installe la police nouvelle. La police est là, il ne reste qu'à empiler le mot 'Postscript' (9) et à en mesurer la largeur.

L'ordre *moveto* (10) (équivalent de HTAB, VTAB sur l'Apple //) nécessite deux arguments, x et y. La page ayant 792 points sur 612, la ligne se charge de centrer le titre.

Suit alors (lignes 11 à 22) la boucle d'impression style FOR I = 1 TO LEN (A$) : PRINT MID$ (A$,I,1); : NEXT, exprimé ici de la façon suivante :

*O 1 L {procédure} for.*

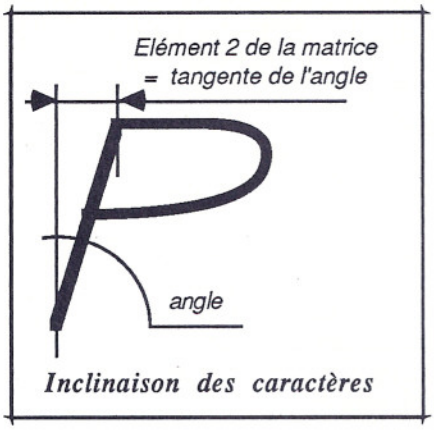
0 est la valeur de départ, 1 l'incrément, L le nombre de caractères -1.

Procédure sera l'ensemble des ordres de recherche du caractère à imprimer, de modification de taille/inclinaison et d'impression.

Notons ici qu'arbitrairement les caractères auront une inclinaison variant de -20° à +20° et leur hauteur variera en suivant une parabole.

*Numcarac* (14) est incrémenté pour suivre le numéro de la lettre en cours.

Ligne 15, l'élément 2 de la matrice nécessaire à makefont, élément qui représente l'inclinaison, est défini ainsi : tangente (numéro de caractère - demilong * 5) * hauteur. Notons que PostScript étant dépourvu de l'ordre TAN, il faut utiliser SIN/COS.



*Inclinaison des caractères*

L'ordre *put* installe la nouvelle inclinaison dans la matrice en élément 2 (2 a été empilé quelques instructions plus tôt).

Même travail (16) sur la hauteur du caractère définie comme suit : (numéro de la lettre - demi-longueur) au carré * 2,5 + 35, 35 étant la hauteur minimale. Les paramètres sont maintenant à jour, on installe la police (17).

Dans la pile, il y a l'indice de boucle qu'on a dupliqué plus haut, on empile chaîne et l'ordre *get* extrait le i$^{ème}$ caractère (18). On met ce caractère dans l'élément 0 de la chaîne tampon *carac* (19).
Il ne reste plus qu'à imprimer (20) cette lettre par l'ordre *show*. Ici on a utilisé l'ordre *ashow* qui nécessite deux paramètres : les valeurs x et y qui sont ajoutées à la position courante avant impression. Dans le cas présent, on demande que les caractères soient un peu plus serrés qu'en standard, sans modification verticale. Cet ordre permet lors des justifications totales de régler l'espace inter-lettre.
La boucle *for* s'exécute ainsi jusqu'à la fin de *chaine*.

Pour imprimer le temps de calcul, on opte pour la police Helvetica (23) en dimension 9 points, *scalefont* se chargeant de mettre à jour la matrice.

Maintenant (26) on empile la valeur actuelle du compteur interne, juste au-dessus de celle empilée ligne 1, on échange les deux valeurs, on soustrait et dans la pile il reste le temps d'exécution. *string* crée une chaîne de caractères vide et *cvs* (convert to string) convertit le temps de calcul en une chaîne imprimable par *show*.
Ultime étape, impression physique par *showpage*.

## Programme SIN

Cet autre exemple a pour objet le dessin d'une courbe et l'impression d'un petit texte.
La courbe définie entre les lignes 4 et 10 est de la forme x = sin (α) * α et y = cos (α) avec α variant de 0 à 360° par pas de 2°. Un test (7) est effectué pour modifier la courbe selon que la valeur du sinus calculé :
*sinus .2 ge {si oui} {si non} ifelse* ce qui se traduit en Basic par *IF SINUS >= .2 THEN si oui ELSE si non*.
Le *lineto* (8) nécessite deux arguments, position horizontale et verticale, pour dessiner un segment de droite reliant ce point à la position courante.
La courbe est maintenant au point (11), on sauvegarde l'état graphique pour la garder en mémoire, on définit un gris clair (*.95 setgray*), on applique le grisé et on restitue l'état graphique antérieur.
Pour laisser la place au futur pointillé, on dessine un trait en blanc (12) d'une largeur de 3 et de couleur 1 (blanc). *stroke* dessine ce trait.
Par *setdash*, il est possible de réalisé toute sorte de pointillé, ici ce sera 20 points noirs, 2 blancs, 2 noirs, 2 blancs. *stroke* réalise le tracé.
Lignes 14 à 21 les chaînes de caractères seront imprimées dix fois, décalées, dans un gris de plus en plus foncé. Dans la définition des chaînes, \( et \) indique à PostScript que la parenthèse n'est pas un délimiteur de chaîne mais un caractère à imprimer.

s'mo9Pom's
s'mo9Pom's

### Comment transmettre Postscript à la LaserWriter ?

**La LaserWriter**

*Elle doit être réglée sur 9600.*

**L'Apple //**

*Carte série configurée comme pour l'ImageWriter, le fichier sera chargé par un traitement de texte et imprimé avec. S'il s'agit d'AppleWriter, ne pas oublier de supprimer le paramètre de soulignement (ps).*

**Le Macintosh**

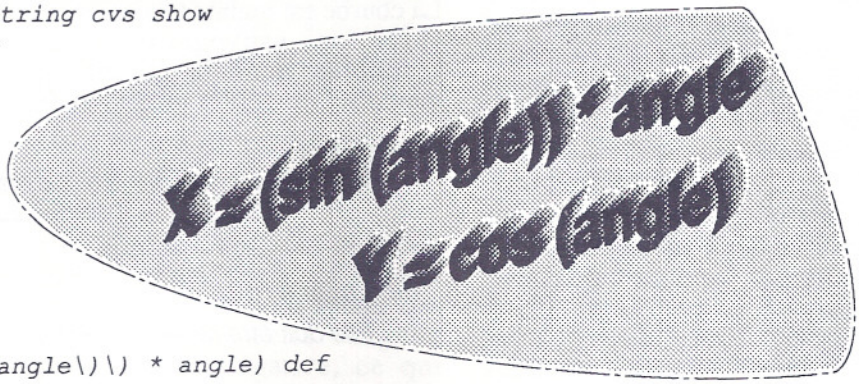
*La LaserWriter sera connectée exactement comme l'ImageWriter, son connecteur 25 broches relié à la sortie imprimante du Mac. Les traitements de texte commandant les imprimantes en mode graphique, le plus simple consiste à transmettre le PostScript (créé sous MacWrite, sauvegardé sous format 'texte seul') à l'aide du programme Basic suivant :*

```
f$=FILES$(1)
OPEN"I",1,f$
LPRINT INPUT$(LOF(1),1)
CLOSE
```

## Programme 'TITRE'

*NB : les numéros de lignes ne font pas partie du programme, ils ne servent qu'aux remarques du texte.*

```
 1  usertime
 2  /police /Times-Roman findfont def
 3  /chaine (Postscript) def
 4  /matrice [40 0 -4 90 0 0] def
 5  /carac 1 string def
 6  /numcarac 0 def
 7  /demilong chaine length 1 add 2 div def
 8  police matrice makefont setfont
 9  chaine stringwidth exch
10  612 exch sub 2 div exch 792 exch sub 2 div moveto
11  0 1 chaine length 1 sub
12    {
13       dup
14       /numcarac numcarac 1 add def
15       /matrice matrice 2 numcarac demilong sub 5 mul dup sin
                 exch cos div 75 mul put
16       matrice 3 35 2.5 2 numcarac demilong sub exch exp mul add put def
17       police matrice makefont setfont
18       chaine exch get
19       carac exch 0 exch put
20       -2 0 carac ashow
21    }
22    for
23  /Helvetica findfont 9 scalefont setfont
24  300 340 moveto
25  (Calcul : ) show
26  usertime exch sub 15 string cvs show
27  ( millisecondes.) show
28  showpage
```




## Programme 'SIN'

```
 1  /chaine1 (X = \(sin \(angle\)\) * angle) def
 2  /chaine2 (Y = cos \(angle\)) def
 3  350 370 moveto
 4  0 2 360
 5  {
 6       dup dup dup
 7       sin dup .2 ge { 2.5 div .3 add } { 2 mul } ifelse mul 2 div 350 add
 8       exch cos 70 mul 300 add lineto
 9  }
10  for
11  gsave .95 setgray fill grestore
12  gsave 3 setlinewidth 1 setgray stroke grestore
13  [20 2 2 2] 0 setdash 0 setlinewidth stroke
14  /Helvetica-Bold findfont [20 4.2 0 25 0 0] makefont setfont
15  1 -.1 0
16  {
17       dup dup dup dup
18       setgray 10 mul dup 130 add exch 285 add moveto chaine1 show
19       setgray 10 mul dup 200 add exch 260 add moveto chaine2 show
20  }
21  for
22  showpage
```

# Un utilitaire d'impression Pascal : Printer

Philippe Magnien

Ce programme a pour principal objet de de débiter les listes des programmes Pascal en tranches d'une page, plus commodes à manipuler que les accordéons continus. En prime, il indique en bas de chaque page la date, repechée dans le système, ce qui permet de repérer sans problème l'édition la plus récente d'une liste. Enfin, les Apple DMP et ImageWriter disposant des caractères français et anglais, il permet de disposer des crochets et accolades en même temps que des caractères accentués. Ces caractères ayant les mêmes code ASCII, l'astuce consiste à considérer qu'un code de 125, par exemple, correspond à un ']' s'il se trouve entre deux apostrophes, et à un crochet fermant dans le cas contraire. Il en résulte que, sur la liste, les codes correspondants seront toujours interprétés de la même façon, suivant qu'ils se trouvent dans la partie programme proprement dite ou dans la partie texte.

Quelques commentaires sur les différentes procédures :

## Datation

Cette *procedure* va rechercher la date dans le système. L'adresse -18432 correspond au Pascal 1.2. Pour le Pascal 1.1, il faut remplacer cette adresse par -21992.

## Initialisation

Cette procédure met en place un certain nombre de constantes qui seront utilisées dans le corps du programme. En particulier, elle initialise la chaîne de caractères "CROCHET" qui sera ensuite utilisée pour provoquer le changement de jeu de caractères lors de la reconnaissance des codes ASCII correspondants. Le

## Programme 'PRINTER.TEXT'

```
program PRINTER;

  const NBL = 58; { pour des pages de 12 pouces; mettre 52 pour 11 pouces }

  type date = packed record
                MOIS : 0..12;
                JOUR : 0..31;
                AN   : 0..99
              end;
      SETOFCHAR = set of char;

  var FICHIER : text;
      AUJOURDHUI :date;
      ENTREE : boolean;
      TAB, I, NUMERO : integer;
      CROCHET,TITRE : string;
      LIGNE : string[160];

  FUNCTION GETCHAR(OKSET : SETOFCHAR): CHAR;

      VAR CH:CHAR;
          GOOD:BOOLEAN;

      BEGIN
        REPEAT
          READ(KEYBOARD,CH);
          IF EOLN(KEYBOARD) THEN CH:=CHR(13);
          GOOD:=CH IN OKSET;
          IF NOT GOOD THEN WRITE(CHR(7))
                      ELSE IF CH IN[' '..'è'] THEN WRITE (CH);
        UNTIL GOOD;
        GETCHAR:=CH
      END;

PROCEDURE GETSTRING(MAXLEN:INTEGER;VAR CHAINE:STRING);

    VAR S : STRING;
        S1 : STRING[1];
        OKSET :SETOFCHAR;

    BEGIN
      S :='';
      S1 :=' ';
      OKSET := [' '..'è'];
      REPEAT
        IF LENGTH (S) = 0
           THEN S1[1] := GETCHAR ( OKSET +[ CHR(13)])
           ELSE IF LENGTH(S) = MAXLEN
                   THEN S1[1] := GETCHAR([CHR(13), CHR(8)])
                   ELSE S1[1] := GETCHAR(OKSET+[CHR(13),CHR(8)]);
                   IF S1[1] IN OKSET THEN S := CONCAT(S,S1)
                         ELSE IF S1[1] =CHR(8)
                                 THEN BEGIN
                                   WRITE( CHR(8),' ',CHR(8));
                                   DELETE(S,LENGTH(S),1)
                                 END;
      UNTIL S1[1] = CHR(13);
      CHAINE:= S;
    END;

  procedure DATATION; {'va chercher la date dans le systeme. cf PEEKs '}
                      {'et POKEs en pascal de J. Pino, POM''S n°12     '}

    type biface = record case boolean of
```

**TurboPascal**

*A la suite de la présentation du TurboPascal (Pom's 23), FRACIEL nous demande de préciser son changement de raison sociale : 'BORLAND FRACIEL' et sa nouvelle adresse : 78, rue de Turbigo, 75003 Paris, tél (1)42 72 25 19.*

programme suppose, pour fonctionner correctement que le jeu de caractères sélectionné au moment où le programme est lancé est le jeu français. C'est toujours le cas en ce qui me concerne, les caractères français étant affichés sur les interrupteurs de l'imprimante. Dans le cas contraire, il convient de rajouter une ligne au programme principal :

```
write(copy(CROCHET,1,4),cop
y(CROCHET,6,4));
```

après l'initialisation de l'imprimante pour se ramener au cas prédédent.

## Traduire

Cette procédure remplace dans la partie programme, les caractères 'é','§','è','°' par la chaîne CROCHET, qui a pour effet de provoquer le passage aux caractères anglo-saxons avant l'envoi du caractère considéré, et le retour aux caractères français ensuite.

### *Comment faire ?*

*Pour lister un programme Pascal, vous devrez faire :*

```
X(ecute
PRINTER
```

*puis vous devrez donner le nom du programme à lister éventuellement précédé de #4: ou #5:.*
*Le programme est conçu pour une imprimante réglée sur 72 lignes par page.*

### *Rendons à Gisèle...*

*...Perreault le mérite qui lui revient : dans Pom's 23, son nom avait été omis en tête de l'article 'formateur de listing' au côté de Sylvie Gallet. Gisèle Perreault nous pardonnera certainement.*

```
                    true : (ADRESSE:integer);
                    false : (CONTENU:^date)
                  end;
    var PEEKPOKET : biface;
        RELAIS : date;

    begin
      PEEKPOKET.ADRESSE := -18342;
      AUJOURDHUI := PEEKPOKET.CONTENU^;
    end;
    procedure INITIALISATION;

      var AUTRETITRE : string;

    begin
      writeln(chr(12));gotoxy(27,0);writeln('****************************');
                       gotoxy(27,1);writeln('*                          *');
                       gotoxy(27,2);writeln('* Utilitaire d''impression *');
                       gotoxy(27,3);writeln('*                          *');
                       gotoxy(27,4);writeln('****************************');
      AUTRETITRE := '';
      ENTREE := true;
      NUMERO := 0;
      LIGNE := '';
      CROCHET := ' Z    D  '; {'crée la séquence d'escape "EscZ60(car)EscD60" }
      CROCHET[1] := chr(27);  {'qui, par remplacement systématique dans les  '}
      CROCHET[3] := chr(6);   {'lignes envoyées à l'imprimante permettra     }
      CROCHET[4] := chr(0);   { d'obtenir les caracteres '} { [ ] ' dans la  '}
      CROCHET[6] := chr(27);  {'partie non texte du programme  }
      CROCHET[8] := chr(6);
      CROCHET[9] := chr(0);
      DATATION;
      writeln;writeln('Nom du fichier à imprimer ? ');
      getstring(40,TITRE);
      writeln;
      (*$I-*)   {'évite le plantage sur une erreur d''entrée sortie'}
      reset(FICHIER,TITRE);
      if ioresult <> 0     {'vérifie la présence du fichier sur la disquette'}
        then begin
               AUTRETITRE := concat(TITRE,'.TEXT'); {'essaye en rajoutant '}
               reset(FICHIER,AUTRETITRE);           {' .TEXT              '}
               if ioresult <> 0
                 then begin                         {'retour case départ  '}
                        gotoxy(0,6);write(chr(11));
                        writeln('fichier non présent');
                        ENTREE := false
                      end
             end
        (*$I+*)
    end;

  procedure TRADUIRE; {'remplace les caracteres " é è ' § " par la séquence '}
                      { d'escape, sauf dans les zones de texte              '}

    VAR I : integer;
        CAR :char;

    begin
      I :=1;
      while I <= length(LIGNE) do
      begin
        if LIGNE[I] = chr(39) then repeat I := I+1 until (LIGNE[I] = chr(39))
                                                   or (I = length(LIGNE));
           {' saute les zones de texte, sans dépasser la fin de la ligne '}
        if (LIGNE[I] in ['é','è','''','§']) then
          begin
            CROCHET[5] := LIGNE[I]; {' met le car approprié dans CROCHET '}
            LIGNE:= concat(copy(LIGNE,1,I-1),CROCHET
                           ,copy(LIGNE,I+1,length(LIGNE)-I)); {'remplacement'}
            I := I + 8
          end;
        I := I+1
      end
    end;

  begin   { programme principal }
```

## Vous n'avez ni DMP, ni ImageWriter ?

Voici la signification des codes de contrôle utilisés par PRINTER pour commander l'imprimante. Il vous suffit de les adapter à votre imprimante dans la procédure 'TRADUIRE'.

**Esc Z 60**
Jeu de caractères américains
**Esc D 60**
Jeu de caractères français
**Esc E**
Caractères 'Elite', 12 caractères par pouce
**Esc L010**
Marge gauche à 10 caractères
**CTRL N**
Caractères dilatés
**CTRL O**
Caractères non dilatés
**Esc X**
Début de soulignement
**Esc Y**
Fin de soulignement

```
repeat
  INITIALISATION;
  if not ENTREE
    then begin   {'fichier non trouvé'}
           writeln('voulez-vous faire une autre tentative ?');
           ENTREE := GETCHAR(['O','N','o','n']) in ['n','N'];
           if ENTREE then exit(program)
         end
until ENTREE;
TAB := (80-2*length(TITRE)) div 2; { centrage du titre }
close(output);reset(output,'printer:');
write(chr(27),'E',chr(27),'L010'); {' sélectionne les caracteres 12 cpi   '}
                                   {'(EscE) et une marge gauche de 10 car '}
                                   {'(EscL010) ,à modifier suivant gout   '}
                                   {' et imprimante                       '}
writeln(' ':TAB,chr(14),chr(27),'X',TITRE,chr(27),'Y',chr(15));
for I := 1 to 3 do writeln;
while not eof(FICHIER) do
begin
  NUMERO := NUMERO+1;
    if NUMERO mod NBL = 0
           then begin {'impression de la date et du numéro en '}
                      {'bas de chaque page'}
             for I := 1 to 3 do writeln;
             with AUJOURDHUI do write(JOUR:3,MOIS:3,' 19',AN);
             writeln('page':30,(NUMERO div NBL ):3);
             page(output);
             writeln(' ':TAB,chr(14),chr(27),'X',TITRE,chr(27),'Y',chr(15));
             {'impression du titre en haut de chaque page'}
             {'Ctrl-N passage en car dilatés, EscX débute le soulignement '}
             {'EscY le termine, Ctrl-O revient aux caracteres normaux     '}
             for I := 1 to 3 do writeln;
           end;
    readln(FICHIER, LIGNE); {' lecture d''une ligne  '}
    TRADUIRE;               {' remplacement des car, si nécessaire '}
    writeln(LIGNE)          {' envoi à l'imprimante                 '}
  end;  { while }
for I := 1 to (NBL - (NUMERO mod NBL) + 3) do writeln; {'complete la '}
                                                       {'derniere page'}
with AUJOURDHUI do write(JOUR:3,MOIS:3,' 19',AN);
writeln('page':30,(NUMERO div NBL + 1):3);
close(output);reset(output,'console:');
close(FICHIER,lock)
end.
```

### *Vous... Votre Apple... et Pom's*

*Dans le prochain numéro de Pom's, nous publierons les résultats et enseignements du sondage paru dans le numéro de mars. Nous tenons dès aujourd'hui à remercier les centaines de lecteurs qui ont répondu à notre appel. Vos remarques nous permettent de mieux vous connaître et de mieux centrer votre revue. Voici la liste des heureux 'tirés au sort' :*
*Jean-Marie PLACE à Noyelles-les-Vermelles, Jean GOUTELLE à Saint Étienne, Claudine BUG à Paris, Robert GINTZ à Strasbourg, Simon BOSMAN en Belgique, Bruno INVERARDI à Saint Louis, Edmond GRAND à Vendargues, Christophe PRENEL à Épinay sur Seine, Bertrand MÉRIC à Cadaujac, Laurent FONTAINE au Mans, Stéphane DEHAN à Caudebec lès Elbeuf, Gérard CUVIER à Pfastatt, Nicolas LEPLEY à Fontaine la Malle, René MARTIN à Cognin et Patrick OLIVE à Paris.*

### *E.P.E.5.0*
*Précisions...*

*E.P.E.5.0 est livré sur une disquette DOS 3.3 (pensons à ceux qui n'ont pas de carte 16Ko). Pour utiliser E.P.E.5.0 sous ProDOS, il faut : formater une disquette ProDOS, puis copier sur celle-ci, à l'aide d''utilitaire système' ou de 'Convert', les fichiers STARTUP, EPEON.PRODOS, EPEPRODOS5.0, DOCEPE.PRODOS et CLAVIER5.0.PRO. Il suffit alors de démarrer sur cette nouvelle disquette.*

## L'heure sans carte : datheur

Bruno Fénart

DATHEUR permet de disposer de la date et de l'heure sans carte horloge et ainsi de pouvoir dater les fichiers avec précision. De plus, si vous disposez d'une carte souris, l'heure sera automatiquement incrémentée.

La carte souris étant intégrée dans l'Apple //c, ce programme est donc particulièrement intéressant pour leurs propriétaires, d'autant que l'absence de "slots" empêche d'y insérer une carte horloge.

- si vous possédez une carte souris et si vous le désirez, les heures et les minutes seront affichées sur la première ligne de l'écran, que ce soit en affichage 40 colonnes ou 80 colonnes. Le ":" séparant les heures et les minutes clignotera à chaque seconde ;
- si votre Apple ne possède pas de carte souris, l'heure ne s'incrémentera pas automatiquement, mais vous pourrez modifier l'heure chaque fois que vous le désirerez en tapant simplement la commande *HEURE* puis en réglant l'heure avec les flèches.

# Mode d'emploi

Après *-DATHEUR* ou *BRUN DATHEUR*, la date et l'heure du système apparaissent ou, par défaut, celles de la dernière utilisation du programme. Les minutes sont en vidéo inverse. En utilisant les flèches ⇑ ou ⇓ (ou les touches ">" et "<" pour un Apple ][+), vous allez respectivement les incrémenter ou les décrémenter. Si de même vous voulez modifier les heures, les jours, les mois ou les années, il suffira que vous utilisiez les flèches ⇐ et ⇒ pour déplacer l'affichage en inverse sur les minutes, les heures, les jours, les mois ou les années suivant ce que vous voulez changer. Pour valider la date et l'heure, appuyez sur la touche RETURN. L'heure et la date seront alors affichées sur la première ligne de l'écran. Le ":" clignotant à chaque seconde indique que l'horloge fonctionne.

L'heure sera ensuite affichée jusqu'à ce que vous supprimiez la protection de la première ligne (par exemple en faisant TEXT ou GR, ou encore en passant de l'affichage 40 colonnes à l'affichage 80). L'heure ne sera plus affichée mais l'horloge continuera à fonctionner. Pour l'afficher à nouveau, il suffira de tapez la commande *HEURE*, ce qui vous permettra éventuellement de régler à nouveau l'heure.

L'heure affichée est assez précise et n'est pas affectée par le déroulement d'un programme, sauf si celui-ci utilise la souris. Cependant, les accès aux disques bloquent les interruptions et arrêtent donc momentanément le fonctionnement de l'horloge, ce qui peut provoquer un retard de quelques minutes sur une demi-journée. Enfin, si vous appuyez sur CTRL-Reset ,vous arrêtez l'horloge. Pour la faire repartir, tapez à nouveau *HEURE*.

Si vous n'avez pas de carte souris, l'heure sera affichée comme expliqué ci-dessus, mais elle n'évoluera pas et le ":" ne clignotera pas. Si vous désirez que vos fichiers soient correctement datés, il vous suffira de mettre l'heure avant de les sauvegarder en utilisant la commande... *HEURE*.

Enfin, si vous voulez récupérer la place occupée par le programme ou arrêter l'horloge, il suffit de tapez la commande *LIBERE*. Celle-ci supprime toutes les commandes extérieures et récupère la place qu'elles occupaient. Vous ne pouvez plus utiliser ni la commande *HEURE* ni *LIBERE*, ni aucune des commandes que vous aviez installées, sauf en relançant l'exécution de DATHEUR ou d'un programme installant une commande extérieure.

Note : les commandes extérieures n'existe pas en DOS 3.3, ce programme ne fonctionne donc qu'en ProDOS. Comme de plus le principal interêt de ce programme est de pouvoir dater ses fichiers et que cette possibilité n'existe pas en DOS 3.3, ceci n'est pas un handicap. Vous pouvez toutefois l'adapter en DOS 3.3 moyennant des modifications importantes de l'interpréteur de commandes. Il faut noter, pour ceux qui souhaiteraient le faire, que les parties les plus complexes (celles concernant la souris) restent utilisables sans modification.

# La technique

Le programme a été conçu de façon très modulaire et chacun des modules peut être réutilisé séparément dans un autre programme.

Les différentes techniques utilisées sont : déplacement d'un programme en code machine, recherche de la carte souris et utilisation quelle que soit la machine ou le slot de connection, ajout d'une commande à l'interpréteur de BASIC.SYSTEM, utilisation des interruptions engendrées par le balayage vertical pour mesurer le temps sans empêcher le fonctionnement d'une autre fonction, etc.

## Commandes extérieures

• L'ajout d'une commande à l'interpréteur de BASIC. SYSTEM a été abordée dans le Pom's 20. Rappelons toutefois que ces commandes sont utilisables aussi bien dans un programme Basic avec la même syntaxe que pour une commande ProDOS (à condition bien sûr que la commande soit installée), que directement au clavier.

Note : DATHEUR possède son propre relogeur qui installe les deux commandes sous BASIC.SYSTEM sans détruire les variables Basic, et ceci même si un ou plusieurs fichiers sont ouverts.

## La souris

• Le deuxième point important concerne la souris. L'utilisation de cette dernière est fort complexe mais est grandement facilitée par l'appel aux sous-programmes fournis dans la ROM de la carte souris. L'emplacement de ces sous-programmes est différent pour un Apple //e et un Apple //c, mais une table contenant leurs adresses a été disposée en début de ROM, permettant ainsi au programmeur de réaliser un programme fonctionnant sur les deux machines. C'est ce que permet le sous-programme SOURIS.

• DATHEUR utilise la souris d'une manière particulière, aussi ne détaillerons nous pas l'utilisation de la souris elle-même. Nous ne parlerons ici que des différents modes d'interruptions, interruptions génerées par la souris et qui permettent à l'horloge de fonctionner à une fréquence de 50 Hz. Sans trop entrer dans les détails, il faut savoir que pour que l'on puisse déterminer sa position, la souris génère des impulsions chaque fois qu'on la déplace. En plus de ces interruptions dues aux déplacements, deux autres sources d'interruptions existent ; celles provoquées par la pression du bouton, et

celles qui interviennent à chaque rafraîchissement de l'écran (VBL), ce qui se produit 60 fois par seconde (50 fois pour l'Apple//c). Le programmeur peut choisir d'utiliser chacun de ces modes d'interruptions, soit ensemble, soit séparément. Pour DATHEUR, seule la fréquence de 60 Hz (ou 50 Hz suivant la machine) nous intéresse, nous n'utiliserons donc que les interruptions VBL.

Remarque : sur les modèles américains de l'Apple //c le balayage se fait à une fréquence de 60Hz . C'est cette valeur qui est indiquée dans la documentation Apple, qui ne parle pas des modèles //c français pour lesquels la fréquence est de 50 Hz. Pour l'Apple //c, la précision semble très bonne (de l'ordre de un pour mille). Pour l'Apple //e, elle est un peu inférieure (de l'ordre de cinq pour mille). Vous pouvez affiner le réglage de l'horloge en modifiant la ligne 110 (octet $6007) pour l'Apple //c ou la ligne 113 (octet $600D) pour l'Apple //e ou ][+.

Attention : sur l'Apple //c le vecteur d'interruption pointant en $C803 (NEWIRQ), il est toujours accessible (la zone entre $C100 et $CFFF n'est pas commutable). Comme d'autre part ProDOS se charge d'initialiser les adresses $FFFE et $FFFF en carte langage et en mémoire auxiliaire, il n'y a aucun problème à utiliser un programme en mémoire auxiliaire avec l'Apple //c. Il n'en va pas de même sur un Apple //e . En effet, le vecteur d'interruption pointant en $FA40 (IRQ) n'est pas accessible d'un programme en mémoire auxiliaire. Par contre, et heureusement, ProDOS a pour son usage 'personnel' une routine de prise en charge d'interruption en carte langage.

# Source 'DATHEUR.SCE0'

### Assembleur ProCODE

```
 0  ********************************************
 1  *                                          *
 2  * DATHEUR  V. 2.0   (C) Bruno Fénart  1985 *
 3  *                                          *
 4  * Création:       Juillet  1985            *
 5  * Modification:  Décembre 1985             *
 6  *                                          *
 7  ********************************************
 8
 9  * Assembleur : ProCODE
10
11  * L'exécution de DATHEUR permet d'ajouter
12  * deux commandes supplémentaires à BASIC.SYSTEM :
13  *
14  * HEURE   - Datation et mise à l'heure du système
15  * De plus, s'il y a une carte souris, le temps est
16  * mesuré grâce à des interruptions de 50 ou de 60 Hz
17  * LIBERE  - déconnecte toutes les commandes
18  * additionnelles, y compris HEURE et LIBERE
19
20          ORG  $6000
21
22  *    Variables en page 0
23  *    -------------------
24
25  WNDTOP   EQU  $22        ;Haut de la fenêtre écran
26  LENGTH   EQU  $2F        ;Longueur de l'instruction
27  PCL      EQU  $3A        ;Adresse de la ligne désassemblée
28  ACC      EQU  $45        ;Acc. après une interruption
29  ADR      EQU  $FE        ;Vecteur d'indirection temporaire
30
31  *   Constantes souris
32  *   -----------------
33
34  INITMOUSE EQU $19        ;Initialisation de la souris
35  SETMOUSE EQU  $12        ;Active le mode des interruptions
36  SERVEMOUSE EQU $13       ;Origine de l'interruption ?
37
38  *   Variables non en page 0
39  *   -----------------------
40
41  IRQLOC   EQU  $3FE       ;Vecteur de l'interruption IRQ
42  LINE1    EQU  $400       ;Adresse du début de l'écran
43
44  *  Sous-programmes du moniteur et de l'interpréteur Basic
45  *  ------------------------------------------------------
46
47  INSDSP2  EQU  $F88C      ;Décode la ligne (PCL) si X=0
48  OUTDO    EQU  $DB5C      ;Affiche le code ASCII de Acc.
49  LINPRT   EQU  $ED24      ;Affiche l'entier X,A  (A = MSB)
50  NORMAL   EQU  $F273      ;Passage en mode normal
51  INVERSE  EQU  $F277      ;Passage en mode inverse
52  UP       EQU  $FC1A      ;Remontée éventuelle du curseur
53  HOME     EQU  $FC58      ;Effacage de l'écran
54  TEXT     EQU  $F399      ;Passage écran texte total
55  RDKEY    EQU  $FD0C      ;Attend une touche chargée dans A
56  CROUT    EQU  $FD8E      ;Envoie un retour chariot
58
59  *   Sous-programmes en page globale BASIC.SYSTEM  (BI)
60  *   -------------------------------------------------
61
62  GOSYSTEM EQU  $BE70      ;Exécute une commande ProDOS
63  BADCALL  EQU  $BE8D      ;Code erreur MLI --> code BI
64  ERROUT   EQU  $BE09      ;Affiche l'erreur (code BI en A)
65
66  GETBUFR  EQU  $BEF5      ;Réserve (A) page(s) sous le BI
67  FREEBUFR EQU  $BEF8      ;Libère buffer obtenu par GETBUFR
68
69  *   Variables en page globale BASIC.SYSTEM
70  *   --------------------------------------
71
72  EXTRNCMD EQU  $BE06      ;JMP vers interpréteur extérieur
73  XTRNADDR EQU  $BE50      ;Adresse d'une commande extérieure
74  XLEN     EQU  $BE52      ;Longueur-1 du nom de la commande
75  XCNUM    EQU  $BE53      ;Numéro de la commande
76  PBITS    EQU  $BE54      ;Paramètres optionnels autorisés
77  VPATH1   EQU  $BE6C      ;Vecteur vers la commande entrée
78  XRETURN  EQU  $BE9E      ;RTS connu en page globale du BI
79
80  SSGINFO  EQU  $BEB4      ;Paramètres de GET/SET FILE INFO
81  FIMDATE  EQU  $BEBE      ;Date du dernier fichier utilisé
82
83  *   Variables et sous-programmes en page globale ProDOS
84  *   ---------------------------------------------------
85
86  DATE     EQU  $BF90      ;Date ProDOS
87  MACHID   EQU  $BF98      ;Identificateur de la machine
88  IRQENT   EQU  $BFEB      ;Vecteur entrée d'interruption
89
90  SETINFO  EQU  $C3        ;Code de SET FILE INFO
91  SETCOUNT EQU  $07        ;Nombre de paramètre pour SETINFO
92
93  *   Commutateurs et indicateurs
94  *   ---------------------------
95
96  RD80VID  EQU  $C01F      ;Indic. affichage 80 colonnes
97  TXTPAGE1 EQU  $C054      ;Affichage page 1 et mémoire P
98  TXTPAGE2 EQU  $C055      ;Accès mémoire X si RD80STORE
99          PUT  DATHEUR.SCE1     ;inclut le relogeur
100         PUT  DATHEUR.SCE2     ;inclut les commandes
```

# Source 'DATHEUR.SCE1'

```
 0  * 1' partie: Relogeur
 1  * ===================
 2
 3  ************************************************
 4  *                                              *
 5  * Installation de la commande personnalisée    *
 6  * entre BASIC.SYSTEM (BI) et ses buffers       *
 7  * Recherche s'il y a une carte souris, et si oui *
 8  * détermine le numéro du slot où elle se trouve *
 9  * puis lance l'exécution de la commande HEURE  *
10  *                                              *
```

*Ce source, sur la disquette d'accompagnement, est sauvegardé en format TEXT pour permettre l'édition avec un autre assembleur.*

```
11  ************************************************
12
13          JSR  TROUSOU     ;Recherche le slot de la souris
14          LDA  MACHID      ;Test si la machine est un //c
15          LDX  #25         ;Si c'est un //c alors 50 Hz
16          CMP  #$BA
17          BEQ  $IIc
18          LDX  #30         ;sinon 60 Hz
19  $IIc    STX  MAXHZ
20
21          LDX  #$00        ;Si II ou II+
22          AND  #$88        ;Isole bit 7 et 3 de MACHID
23          BEQ  $IIplus
24          LDX  #$FF        ;Si //e ou //c
25  $IIplus STX  APPLEID
26
27          LDA  EXTRNCMD+2 ;Préserve la commande précédente
28          STA  PRECMD+2
29          LDA  EXTRNCMD+1
30          STA  PRECMD+1
31
32          LDA  #>NBPAG     ;Réserve autant de pages qu'en
33          JSR  GETBUFR     ;occupe le programme à reloger
34          CMP  #$0C        ;Si pas d'erreur (A)= adr. buffer
35          BNE  $gothem     ;sinon (A) = code d'erreur BI
36          JMP  ERROUT      ;Affiche "NO BUFFERS AVAILABLE"
37
38  $gothem STA  EXTRNCMD+2 ;Installe la commande HEURE
39          STA  ADR+1       ;Initialise adresse de transfert
40          LDA  #0
41          STA  EXTRNCMD+1
42          STA  ADR
43          LDA  #>DEBUT     ;Adresse du programme à reloger
44          STA  PCL+1
45          LDA  #<DEBUT
46          STA  PCL
47          JSR  RELOGE      ;Reloge le programme sous le BI
48          JSR  CMDH        ;Exécute la commande HEURE
49          LDA  #SETCOUNT   ;sinon modifie les informations
50          STA  SSGINFO     ;du fichier DATHEUR pour mettre
51          LDA  #SETINFO    ;la date de dernière utilisation
52          JSR  GOSYSTEM
53          BCC  $noerr      ;S'il n'y a pas d'erreur,
54          JSR  BADCALL     ;Convertit le code MLI en code BI
55          JMP  ERROUT      ;Affiche l'erreur et fin
56
57  $noerr  CLC              ;Pas d'erreur (sinon code dans A)
58          RTS              ;This is the end my friend
59
60
61  *   Relogeur
62  *   --------
63
64  * Reloge le programme de son ancienne adresse = DEBUT
65  * vers son implantation permanente sous le BI = (EXTRNCMD)
66  * Les deux adresses doivent être des multiples de 256
67  * Le nombre de pages transférées est égal à NBPAG
68  * La zone des données est séparée par DFB $00 (= BRK)
69  * Le programme prend en compte la structure d'appel au MLI
70
71  RELOGE  LDX  #0          ;Indispensable pour INSDSP2
72          LDY  #$FF        ;Pour le transfert des données
73          LDA  (PCL,X)     ;L'instruction est égale à BRK
74          BEQ  $trans      ;alors début des données
75          JSR  INSDSP2     ;Décode une instruction et
76          LDY  LENGTH      ;détermine sa longueur
77          CPY  #2
78          BNE  $trans      ;Longueur < 3 : Opérande inchangé
79          LDA  (PCL),Y     ;L'instruction est-elle JSR $BF00
80          CMP  #$BF
81          BNE  $nomli
82          DEY
83          LDA  (PCL),Y
84          CMP  #$00
85          BNE  $nomli
86          DEY
87          LDA  (PCL),Y
88          LDY  #5          ;Intruction sur 6 octets du MLI
89          CMP  #$20
90          BEQ  $mli
91  $nomli  LDY  #2          ;Intruction sur trois octets
92  $mli    LDA  (PCL),Y     ;Sinon poids fort de l'opérande
93          SEC
94          SBC  #>DEBUT     ;comparé avec l'adresse de début
95          BCC  $trans      ;Opérande < adresse de début
96          CMP  #>NBPAG
97          BCS  $trans      ;ou opérande > adresse de fin
98          ADC  EXTRNCMD+2 ;Sinon calcule l'octet de poids
99          STA  (PCL),Y     ;fort de l'adresse translatée
100
101 $trans  LDX  #0
102         INY              ;Longueur réelle
103 $boucle LDA  (PCL,X)     ;Transfert un octet
104         STA  (ADR,X)     ;vers sa nouvelle adresse
105         INC  PCL         ;Calcul des adresses suivantes
106         INC  ADR
107         BNE  $nocarry
108         INC  PCL+1
109         INC  ADR+1
110         LDA  #>DEBUT+NBPAG;Numéro de la page finale + 1
111         CMP  PCL+1       ;Fin du programme si on a
112         BEQ  $fin        ;transféré les NBPAG pages
113 $nocarry DEY             ;Transfert 1 à 3 octets suivant
114         BMI  $trans      ;Transfert en continue (données)
115         BNE  $boucle     ;la longueur de l'instruction
116         BEQ  RELOGE      ;Instruction suivante
117
118 $fin    RTS
119
120
121 *   Recherche la souris
122 *   -------------------
123
124 * Détermine s'il y a une carte souris et, si oui,
125 * recherche le slot où elle est connectée. Le résultat est
126 * mis dans MOUSLOT. Il vaut zéro s'il n'y a pas de souris
127
128 TROUSOU LDA  #$C8         ;Détermination du slot souris
129         STA  ADR+1
130         LDA  #$00
131         STA  ADR
132         LDX  #7          ;Recherche parmi les 7 slots
133 $suivant DEC ADR+1       ;Slot suivant
134         DEX
135         BMI  $nomouse    ;Souris dans aucuns des slots
136
137         LDY  #$05        ;Signature d'un périphérique
138         LDA  #$38
139         CMP  (ADR),Y
140         BNE  $suivant    ;sinon, slot suivant
141
142         LDY  #$07        ;Idem
143         LDA  #$18
144         CMP  (ADR),Y
145         BNE  $suivant
146
147         LDY  #$0B        ;Idem
148         LDA  #$01
149         CMP  (ADR),Y
150         BNE  $suivant
151
152         LDY  #$0C        ;Signature d'un joystick
153         LDA  #$20        ;ou équivalent (paddle,
154         CMP  (ADR),Y     ;souris, etc.)
155         BNE  $suivant    ;sinon, slot suivant
156
157         LDY  #$FB        ;Signature de la souris
158         LDA  #$D6
159         CMP  (ADR),Y
160         BNE  $suivant    ;sinon, slot suivant
161
162         LDA  ADR+1       ;Poids fort du slot souris
163         STA  MOUSLOT
164         RTS
165 $nomouse LDA  #0         ;S'il n'y a pas de carte souris
166         STA  MOUSLOT     ;alors MOUSLOT = 0
167         RTS
```

## Source 'DATHEUR.SCE2'

```
0   * 2' partie: Code relogeable permanent
1   *=====================================
2
3           DS   $6100-*     ;Le début doit être aligné
4
5   ********************************************************
6   *                                                      *
7   *     Interpréteur des commandes HEURE et LIBERE       *
8   *   Si aucune des deux commandes n'est reconnue, on    *
9   * retourne à l'adresse d'une éventuelle autre commande *
10  *                                                      *
```

```
11  ******************************************************
12
13  DEBUT    CLD              ;Non indispensable mais prudence
14           LDA  VPATH1      ;Adresse-1 de la commande entrée
15           STA  ADR         ;mise dans un pointeur temporaire
16           LDA  VPATH1+1
17           STA  ADR+1
18           LDX  #$FF        ;Initialise sur la lière commande
19  $cherche LDY  #$00        ;Pointe l' caractère de l'entrée
20  $compare INX              ;Caractère suivant
21           INY
22           LDA  CMDLIST,X   ;Charge une lettre de la commande
23           BPL  $trouvé     ;Comparaison complète
24           AND  #$7F        ;Conversion ASCII positif
25           CMP  (ADR),Y     ;Comparaison avec l'entrée
26           BEQ  $compare    ;Majuscules identiques: suivante
27           EOR  #$20        ;sinon compare avec une minuscule
28           CMP  (ADR),Y     ;Recomparaison
29           BEQ  $compare    ;Minuscules identiques: suivante
30  $autre   INX              ;Recherche un pointeur commande
31           LDA  CMDLIST,X
32           BMI  $autre      ;Un caractère d'une commande ?
33           BPL  $cherche    ;Va tester la commande suivante
34
35  $trouvé  BEQ  $synerr     ;Pointeur nul = commande inconnu
36           TAX              ;sinon il est mis comme indice
37           LDA  (ADR),Y     ;Le caractère suivant l'entrée
38           CMP  #$0D        ;doit être un retour chariot
39           BNE  $synerr     ;sinon erreur de syntaxe
40           LDA  CMDVECT,X   ;Charge l'adresse de la commande
41           STA  XTRNADDR
42           LDA  CMDVECT+1,X;Idem poids fort
43           STA  XTRNADDR+1
44           DEY              ;Mémorise la longueur - 1
45           STY  XLEN
46           LDA  #$00
47           STA  XCNUM       ;Numéro d'une commande extérieure
48           STA  PBITS+1     ;Aucun paramètres autorisés
49           STA  PBITS
50           CLC              ;La commande a été trouvé
51           RTS
52  $synerr  SEC              ;Erreur de syntaxe
53  PRECMD   JMP  $0000       ;Saut à la commande précédente
54
55  *   Vecteurs d'indirection vers les commandes
56  *   ----------------------------------------
57
58  CMDVECT  EQU  *           ;Vecteurs des commandes
59  CMDS     JMP  LIBERE      ;Les JMP sont indispensables
60  CMDH     JMP  HEURE       ;pour permettre le relogement
61
62
63  ******************************************************
64  *                                                    *
65  *                Commande LIBERE                     *
66  *      Supprime toutes les commandes installées par  *
67  * l'utilisateur, libère tous les buffers et désactive, *
68  * s'il y a lieu, les interruptions de la souris      *
69  *                                                    *
70  ******************************************************
71
72  LIBERE   LDA  #>XRETURN   ;Renvoie le pointeur de commande
73           STA  EXTRNCMD+2 ;extérieure sur un RTS connu
74           LDA  #<XRETURN
75           STA  EXTRNCMD+1
76
77           LDA  MOUSLOT     ;Souris ?
78           BEQ  $noirq      ;Pas touche à mon IRQ
79           SEI
80           LDA  #0          ;Désactive la souris
81           LDY  #SETMOUSE
82           JSR  SOURIS
83           LDA  #>IRQENT    ;Revectorise le vecteur IRQ
84           STA  IRQLOC+1
85           LDA  #<IRQENT
86           STA  IRQLOC
87           CLI
88
89  $noirq   JMP  FREEBUFR    ;Libère tous les buffers et fin
90
91
92  ******************************************************
93  *                                                    *
94  *                Commande HEURE                      *
95  *          datation et mise à l'heure du système     *
96  *                                                    *
97  *   Pour ceux qui veulent connaître le statut de la  *
98  *   commande HEURE, cf J.B. POQUELIN in Oeuvres complètes *
99  *                                                    *
100 ******************************************************
101
102 * Met à jour la date du système à l'aide des touches
103 * montante pour incrémenter, descendante pour décrementer,
104 * <- et -> pour passer alternativement des jours aux mois,
105 * ou aux années ou aux heures ou aux minutes.
106 * Les mois sont affichés en abréviations de trois lettres.
107 * La date affichée est celle du système ou par défaut
108 * celle de la dernière utilisation du programme
109 * ou encore le 01/JAN/86 00:00.
110
111 HEURE   LDA  #<DATE      ;Date du système
112          STA  ADR
113          LDA  #>DATE
114          STA  ADR+1
115          JSR  TESTIME     ;Conversion et test de la date
116          BCC  $datée      ;Si le système est déjà daté
117          LDA  #<FIMDATE   ;Date du fichier
118          STA  ADR
119          LDA  #>FIMDATE
120          STA  ADR+1
121          JSR  TESTIME     ;One more time
122          BCC  $datée      ;Si le fichier est déjà daté
123          LDA  #86         ;Date par défaut 1 Janvier 86
124          STA  AA
125          LDA  #1
126          STA  MM
127          LDA  #1
128          STA  JJ
129          LDA  #0          ;Heure par défaut 00h00
130          STA  HH
131          STA  MN
132
133 $datée   LDX  #0          ;Initialise TAB à droite
134          STX  TAB
135 $boucle  JSR  AFFICHE     ;Boucle de mise de la date
136          JSR  CLAVIER
137          BCC  $boucle
138
139          JSR  ACTIVBL     ;Active les interruptions VBL
140          JSR  NORMAL      ;Annule un inverse éventuel
141          JSR  TEXT        ;Efface totalement l'écran
142          JSR  HOME        ;et positionne en haut à gauche
143          LDA  #5          ;Déconnecte inverse pour AFFICHE
144          STA  TAB
145          JSR  AFFICHE     ;Affichage définitif
146          LDA  #1          ;Protége la première ligne
147          STA  WNDTOP
148          LDA  #$FF        ;Autorise l'affichage de l'heure
149          STA  AFFLAG
150          JSR  SAUVHEUR    ;Sauvegarde heure et date système
151          CLC              ;Pas d'erreur (sinon code dans A)
152          RTS              ;This is the end my friend
153
154
155 *   Conversion et test de la date
156 *   -----------------------------
157
158 * Conversion de la date sur 4 octets pointés par (ADR)
159 * Test de validité puis mémorisation dans MN,HH,JJ,MM,AA
160
161 TESTIME LDY  #3
162          LDA  (ADR),Y     ;Heure
163          CMP  #24
164          BCC  $hok
165          LDA  #0          ;Si heure fausse, mise à 0
166 $hok     STA  HH
167
168          DEY
169          LDA  (ADR),Y     ;Minute
170          CMP  #60
171          BCC  $mnok
172          LDA  #0          ;Si minute fausse, mise à 0
173 $mnok    STA  MN
174
175          DEY              ;Année
176          LDA  (ADR),Y
177          CMP  #200        ;Non daté si année > 100
178          BCS  $nodate
179          LSR
180          STA  AA
181
182          DEY              ;Mois
183          LDA  (ADR),Y
184          TAX
```

```
185          ROR
186          LSR
187          LSR
188          LSR
189          LSR
190          BEQ  $nodate     ;Non daté si le mois est < 0
191          CMP  #13         ; ou  > 12
192          BCS  $nodate
193          STA  MM
194
195          TXA              ;Jour
196          AND  #$1F
197          BEQ  $nodate     ;Non daté si le jour est = 0
198          STA  JJ
199          CLC
200          RTS
201
202 $nodate  SEC
203          RTS
204
205
206 *   Conversion inverse de la date
207 *   -----------------------------
208
209 *  Conversion de la date JJ,MM,AA et de l'heure HH,MN   *
210 *  puis sauvegarde dans les adresses système et fichier *
211
212 SAUVHEUR LDA  HH
213          STA  DATE+3
214          STA  FIMDATE+3
215          LDA  MN
216          STA  DATE+2
217          STA  FIMDATE+2
218          LDA  MM
219          ASL
220          ASL
221          ASL
222          ASL
223          ASL
224          ORA  JJ
225          STA  DATE
226          STA  FIMDATE
227          LDA  AA
228          ROL
229          STA  DATE+1
230          STA  FIMDATE+1
231          RTS
232
233
234 *   Gestion du clavier
235 *   ------------------
236
237 *  Incréments et décrements suivant les flèches
238 *  Retour avec le carry mis en cas de retour chariot
239
240 CLAVIER  JSR  RDKEY
241          LDY  TAB
242          LDX  ORDRE,Y
243          CMP  #$95        ;Flèche -->
244          BEQ  $droite
245          CMP  #$A0        ;Idem 80 colonnes (sauf //c)
246          BEQ  $droite
247          CMP  #$88        ;Flèche <--
248          BEQ  $gauche
249          CMP  #$8A        ;Flèche descendante
250          BEQ  $décréme
251          CMP  #$BC        ;Signe "<" (II+)
252          BEQ  $décréme
253          CMP  #$8B        ;Flèche montante
254          BEQ  $incréme
255          CMP  #$BE        ;Signe ">" (II+)
256          BEQ  $incréme
257          CMP  #$8D
258          BNE  $retour
259          SEC
260          RTS
261 $retour  CLC
262          RTS
263
264 $gauche  CPY  #4          ;Tabulation vers la gauche
265          INY
266          BCC  $nondroi
267          LDY  #0          ;Retour à droite
268 $nondroi STY  TAB
269          BPL  $retour
270
271 $droite  DEY              ;Tabulation vers la droite
272          BPL  $nongau
273          LDY  #4          ;Retour à gauche
274 $nongau  STY  TAB
275          BPL  $retour
276
277 $incréme INC  DATAB,X     ;Incrémente suivant valeur de TAB
278          LDA  MAX,X       ;Compare avec le maximum
279          CMP  DATAB,X
280          BCS  $retour
281          LDA  MIN,X       ;Revient au minimum
282          STA  DATAB,X
283          DEX              ;Incrémente en chaine
284          BPL  $incréme    ;successivement MN,HH,JJ,MM,AA
285          BMI  $retour
286
287 $décréme LDA  MIN,X       ;Compare avec le minimum
288          CMP  DATAB,X
289          DEC  DATAB,X     ;Décrémente suivant valeur de TAB
290          BCC  $retour
291          LDA  MAX,X       ;Revient au maximum
292          STA  DATAB,X
293          DEX              ;Décrémente en chaîne
294          BPL  $décréme    ;successivement MN,HH,JJ,MM,AA
295          BMI  $retour
296
297
298 *   Affichage de la date
299 *   --------------------
300
301 * Affiche le jour, le mois (en 3 lettres), l'année
302 * et l'heure suivant les valeur de JJ,MM,AA,HH,MN.
303
304 AFFICHE  JSR  CROUT       ;Bord gauche de la fenêtre que
305          JSR  UP          ;l'on soit en 40 ou 80 colonnes
306
307          LDY  #4
308 $boucle  TYA              ;Sauvegarde Y
309          PHA
310          JSR  NORMAL      ;Passage en mode vidéo normal
311
312          LDA  ESPACE,Y    ;Affiche caractère d'espacement
313          JSR  OUTDO
314
315          CPY  TAB         ;Affichage en inverse suivant TAB
316          BNE  $normal
317          JSR  INVERSE
318
319 $normal  LDX  ORDRE,Y     ;Successivement JJ,MM,AA,HH,MN
320          LDA  DATAB,X
321          CPX  #1          ;Si ce n'est mois c'est donc ...
322          BEQ  $affmois
323
324          TAX
325          CPX  #10         ;Si strictement inférieur à 10,
326          BCS  $dizaine
327          LDA  #"0"        ;affiche un zéro en plus
328          JSR  OUTDO
329 $dizaine LDA  #00         ;Affiche X (car A=0)
330          JSR  LINPRT      ;c-à-d successivement JJ,AA,HH,MN
331
332 $resto   PLA              ;Restaure Y
333          TAY
334          DEY
335          BPL  $boucle
336          RTS
337
338 $affmois ASL              ;MM x 3
339          ADC  MM
340          TAX
341          LDY  #3          ;Boucle sur les trois lettres
342 $letrois LDA  MOIS-3,X
343          JSR  OUTDO       ;Affiche un des 3 caractères
344          INX
345          DEY
346          BNE  $letrois
347          BEQ  $resto      ;Suite
348
349
350 *   Activation des interruptions
351 *   ----------------------------
352
353 * Active la souris afin d'autoriser les interruptions VBL
354 * donc une fréquence de branchement vers IRQ50HZ de 50Hz
355
356 ACTIVBL  LDA  MOUSLOT     ;Teste s'il y a une souris
357          BEQ  $noirq      ;sinon pas d'interruption
358          SEI
```

```
359          LDA  $vectirq+1 ;Revectorise l'interruption IRQ
360          STA  IRQLOC
361          LDA  $vectirq+2
362          STA  IRQLOC+1
363          LDY  #INITMOUSE ;Initialise la souris
364          JSR  SOURIS
365          LDA  #8         ;Active la souris et autorise
366          LDY  #SETMOUSE  ;les interruptions VBL
367          JSR  SOURIS
368          CLI
369 $noirq   RTS
370
371 $vectirq JMP  IRQ50HZ    ;Adresse de l'interruption (rel)
372
373 *   Sous-programmes souris
374 *   ----------------------
375
376 * Appelle un sous-programme souris pointé par Y dans la
377 * table de la carte souris (contrairement aux programmes
378 * cette table est à la même place pour un //c et un //e)
379
380 SOURIS  PHA              ;Préserve un éventuel paramètre
381          LDX  #0         ;ADR <-- adresse de la souris
382          STX  ADR        ;(et non son numéro de téléphone)
383          LDX  MOUSLOT    ;$Cn dans X avec n = slot souris
384          STX  ADR+1
385          LDA  (ADR),Y    ;Poids faible du sous-programme
386          STA  ADR        ;dans le vecteur d'indirection
387          TXA             ;Duplique $Cn dans A
388          ASL
389          ASL
390          ASL
391          ASL
392          TAY             ;$n0 dans Y
393          PLA             ;Récupère A
394          JMP  (ADR)      ;Programme souris correspondant
395
396
397 ********************************************************
398 *                                                      *
399 *  IRQ50HZ : Gestion des interruptions dues au balayage *
400 * vertical de l'écran toutes les 20 millisecondes (50Hz) *
401 *   Maintient le système à l'heure et l'affiche sur la  *
402 *   première ligne de l'écran si celle-ci est protégée  *
403 *                                                      *
404 ********************************************************
405
406 IRQ50HZ  SEI
407          LDA  ADR        ;Préserve ADR, Acc, X, Y
408          STA  SAVE
409          LDA  ADR+1
410          STA  SAVE+1
411          LDA  ACC
412          STA  SAVA
413          STX  SAVX
414          STY  SAVY
415          LDA  #0         ;Mode inactif
416          LDY  #SETMOUSE
417          JSR  SOURIS
418          LDY  #SERVEMOUSE;Origine de l'interruption
419          JSR  SOURIS
420          BCC  $notre     ;C'est à nous
421          LDA  SAVE       ;Sinon, récupère ADR, Acc, X, Y
422          STA  ADR
423          LDA  SAVE+1
424          STA  ADR+1
425          LDY  SAVY
426          LDX  SAVX
427          LDA  SAVA
428          JMP  IRQENT     ;C'est pas à nous
429 $notre   LDA  WNDTOP     ;La lière ligne est protégée ?
430          CMP  #1
431          BEQ  $okflag
432          LDA  #0         ;Sinon plus d'affichage
433          STA  AFFLAG
434 $okflag  DEC  HZ
435          BEQ  $seconde   ;25 interruptions (1/2 s.) ?
436          JMP  $finirq    ;(Branchement trop long)
437
438 $seconde LDA  MAXHZ      ;Init. compteur d'interruptions
439          STA  HZ
440          LDA  AFFLAG     ;Test le drapeau d'affichage
441          BEQ  $noclign   ;si = 0 pas de ":" clignotant
442          LDX  #$0D       ;Position du ":" en 40 colonnes
443          BIT  APPLEID    ;II & II+ ou //e & //c ?
444          BPL  $clign40
445          BIT  RD80VID    ;Affichage 80 colonnes ?
446          BPL  $clign40
447          LDX  #$06       ;Position du ":" en 80 colonnes
448 $clign40 LDA  #$1A       ;Fait clignoter le ":"
449          EOR  LINE1,X
450          STA  LINE1,X
451 $noclign DEC  SC
452          BNE  $finirq    ;60 secondes ?
453
454          LDA  #120       ;Compteur des 1/2 secondes
455          STA  SC
456          INC  DATE+2     ;Incrémente les minutes (système)
457          LDA  DATE+2
458          CMP  #60
459          BNE  $montre
460          LDA  #0
461          STA  DATE+2
462          INC  DATE+3     ;Incrémente les heures (système)
463          LDA  DATE+3
464          LDY  #"9"       ;Maximum des unités des heures
465          CMP  #24
466          BNE  $montre
467          LDA  #0
468          STA  DATE+3
469          LDY  #"3"       ;Maximum modifié si HH = 23 heures
470
471 $montre  LDA  AFFLAG     ;Affichage autorisé ?
472          BEQ  $finirq    ;Sinon interruption terminée
473          STY  MAXIRQ+1   ;Mémorise le maximum des heures
474          LDY  #15        ;Pointe sur l'unité des minutes
475 $incréme TYA
476          BIT  APPLEID    ;II & II+ ou //e & //c ?
477          BPL  $cadran
478          BIT  RD80VID    ;Affichage 80 colonnes ?
479          BPL  $cadran
480          LSR             ;Divise par 2 et test de parité
481          BCS  $cadran    ;Si impaire => mémoire principale
482          BIT  TXTPAGE2   ;sinon => mémoire principale
483 $cadran  TAX
484          INC  LINE1,X    ;Incrémente un chiffre suivant X
485          LDA  MAXIRQ-11,Y;Maximum dépassé ?
486          CMP  LINE1,X
487          BCS  $finirq    ;sinon fin
488          LDA  #"0"       ;si oui remise à zéro
489          STA  LINE1,X
490 $double  DEY             ;et pointe le chiffre suivant
491          CPY  #13        ;Pointe sur ":" ?
492          BEQ  $double    ;si oui pointe le chiffre suivant
493          BIT  TXTPAGE1   ;Remet la mémoire principale
494          CPY  #10
495          BNE  $incréme   ;Incrémente en chaîne
496 $finirq  BIT  TXTPAGE1
497          LDA  #8         ;Interrution VBL
498          LDY  #SETMOUSE
499          JSR  SOURIS
500          LDA  SAVE       ;Récupère ADR, Acc, X, Y
501          STA  ADR
502          LDA  SAVE+1
503          STA  ADR+1
504          LDY  SAVY
505          LDX  SAVX
506          LDA  SAVA
507          CLI
508          RTI
509
510
511 ********************************************************
512 *                                                      *
513 *   Zone des variables et constantes des divers modules *
514 *                                                      *
515 ********************************************************
516
517          DFB  0          ;Séparateur
518
519 *   Constantes de l'interpréteur de commandes
520 *   -----------------------------------------
521 * Les caractères sont en ASCII négatif
522 * Les pointeurs sont toujours inférieur à $80
523
524 CMDLIST  EQU  *
525          ASC  "LIBERE"
526          DFB  CMDS-CMDVECT+1;Pointeur et séparateur
527          ASC  "HEURE"
528          DFB  CMDH-CMDVECT+1;Idem
529          DFB  0          ;Séparateur final
530
531 *   Variables du module HEURE
532 *   -------------------------
```

```
533
534  DATAB    EQU  *
535  AA       DS   1          ;Année
536  MM       DS   1          ;Mois
537  JJ       DS   1          ;Jour
538  HH       DS   1          ;Heure
539  MN       DS   1          ;Minute
540  TAB      DS   1          ;Tabulation
541  MOUSLOT  DS   1          ;Slot souris ou sinon = 0
542
543  *   Constantes alphanumériques du module HEURE
544  *   ------------------------------------------
545
546  MOIS     ASC  "JANFEVMAR"
547           ASC  "AVRMAIJUN"
548           ASC  "JULAOUSEP"
549           ASC  "OCTNOVDEC"
550  ESPACE   ASC  ": // "
551  ORDRE    DFB  4,3,0,1,2  ;Ordre mémoire de MN,HH,AA,MM,JJ
552  MAX      DFB  99,12,31,23,59;Maximum de AA,MM,JJ,HH,MN
553  MIN      DFB  0,1,1,0,0  ;Minimum de AA,MM,JJ,HH,MN
554
555
556  *   Variables et constantes du module d'interruption
557  *   ------------------------------------------------
558
559  SAVE     DS   2          ;Sauvegarde des adresses $FE, $FF
560  SAVA     DS   1          ;Idem Accumulateur
561  SAVX     DS   1          ;Idem X
562  SAVY     DS   1          ;Idem Y
563
564  AFFLAG   DS   1          ;Drapeau indicateur d'affichage
565  APPLEID  DS   1          ;Identificateur. $FF si //e & //c
566  MAXHZ    DS   1          ;Fréquence: 60 Hz sauf //c : 50 Hz
567  HZ       DFB  30         ;Compteur d'interruptions
568  SC       DFB  120        ;Compteur des demi-secondes
569  MAXIRQ   DFB  "2","9",0,"5","9";Maximum pour chaque chiffre
570
571  *=====================================================*
572  *          Taille du programme à reloger
573
574  NBPAG    EQU  *-DEBUT+$FF;Nombre de pages à reloger
```

# Récapitulation 'DATHEUR'

*Après avoir saisi ce code sous moniteur, vous le sauvegarderez par BSAVE DATHEUR,A$6000,L$494*

```
6000- 20 BA 60 AD 98 BF A2 19
6008- C9 BA F0 02 A2 1E 8E 8C
6010- 64 A2 00 29 88 F0 02 A2
6018- FF 8E 8B 64 AD 08 BE 8D
6020- 53 61 AD 07 BE 8D 52 61
6028- A9 04 20 F5 BE C9 0C D0
6030- 03 4C 09 BE 8D 08 BE 85
6038- FF A9 00 8D 07 BE 85 FE
6040- A9 61 85 3B A9 00 85 3A
6048- 20 62 60 20 57 61 A9 07
6050- 8D B4 BE A9 C3 20 70 BE
6058- 90 06 20 8D BE 4C 09 BE
6060- 18 60 A2 00 A0 FF A1 3A
6068- F0 31 20 8C F8 A4 2F C0
6070- 02 D0 28 B1 3A C9 BF D0
6078- 10 88 B1 3A C9 00 D0 09
6080- 88 B1 3A A0 05 C9 20 F0
6088- 02 A0 02 B1 3A 38 E9 61
6090- 90 09 C9 04 B0 05 6D 08
6098- BE 91 3A A2 00 C8 A1 3A
60A0- 81 FE E6 3A E6 FE D0 0A
60A8- E6 3B E6 FF A9 65 C5 3B
60B0- F0 07 88 30 E6 D0 E7 F0
60B8- A9 60 A9 C8 85 FF A9 00
60C0- 85 FE A2 07 C6 FF CA 30
60C8- 2E A0 05 A9 38 D1 FE D0
60D0- F3 A0 07 A9 18 D1 FE D0
60D8- EB A0 0B A9 01 D1 FE D0
60E0- E3 A0 0C A9 20 D1 FE D0
60E8- DB A0 FB A9 D6 D1 FE D0
60F0- D3 A5 FF 8D 4C 64 60 A9
60F8- 00 8D 4C 64 60 00 00 00
6100- D8 AD 6C BE 85 FE AD 6D
6108- BE 85 FF A2 FF A0 00 E8
6110- C8 BD 38 64 10 14 29 7F
6118- D1 FE F0 F3 49 20 D1 FE
6120- F0 ED E8 BD 38 64 30 FA
6128- 10 E3 F0 24 AA B1 FE C9
6130- 0D D0 1D BD 54 61 8D 50
6138- BE BD 55 61 8D 51 BE 88
6140- 8C 52 BE A9 00 8D 53 BE
6148- 8D 55 BE 8D 54 BE 18 60
6150- 38 4C 00 00 4C 5A 61 4C
6158- 7F 61 A9 BE 8D 08 BE A9
6160- 9E 8D 07 BE AD 4C 64 F0
6168- 13 78 A9 00 A0 12 20 2E
6170- 63 A9 BF 8D FF 03 A9 EB
6178- 8D FE 03 58 4C F8 BE A9
6180- 90 85 FE A9 BF 85 FF 20
6188- DF 61 90 24 A9 BE 85 FE
6190- A9 BE 85 FF 20 DF 61 90
6198- 17 A9 56 8D 46 64 A9 01
61A0- 8D 47 64 A9 01 8D 48 64
61A8- A9 00 8D 49 64 8D 4A 64
61B0- A2 00 8E 4B 64 20 BE 62
61B8- 20 4F 62 90 F8 20 0B 63
61C0- 20 73 F2 20 99 F3 20 58
61C8- FC A9 05 8D 4B 64 20 BE
61D0- 62 A9 01 85 22 A9 FF 8D
61D8- 8A 64 20 21 62 18 60 A0
61E0- 03 B1 FE C9 18 90 02 A9
61E8- 00 8D 49 64 88 B1 FE C9
61F0- 3C 90 02 A9 00 8D 4A 64
61F8- 88 B1 FE C9 C8 B0 20 4A
6200- 8D 46 64 88 B1 FE AA 6A
6208- 4A 4A 4A 4A F0 11 C9 0D
6210- B0 0D 8D 47 64 8A 29 1F
6218- F0 05 8D 48 64 18 60 38
6220- 60 AD 49 64 8D 93 BF 8D
6228- C1 BE AD 4A 64 8D 92 BF
6230- 8D C0 BE AD 47 64 0A 0A
6238- 0A 0A 0A 0D 48 64 8D 90
6240- BF 8D BE BE AD 46 64 2A
6248- 8D 91 BF 8D BF BE 60 20
6250- 0C FD AC 4B 64 BE 76 64
6258- C9 95 F0 2C C9 A0 F0 28
6260- C9 88 F0 18 C9 8A F0 40
6268- C9 BC F0 3C C9 8B F0 22
6270- C9 BE F0 1E C9 8D D0 02
6278- 38 60 18 60 C0 04 C8 90
6280- 02 A0 00 8C 4B 64 10 F2
6288- 88 10 02 A0 04 8C 4B 64
6290- 10 E8 FE 46 64 BD 7B 64
6298- DD 46 64 B0 DD BD 80 64
62A0- 9D 46 64 CA 10 EC 30 D2
62A8- BD 80 64 DD 46 64 DE 46
62B0- 64 90 C7 BD 7B 64 9D 46
62B8- 64 CA 10 EC 30 BC 20 8E
62C0- FD 20 1A FC A0 04 98 48
62C8- 20 73 F2 B9 71 64 20 5C
62D0- DB CC 4B 64 D0 03 20 77
62D8- F2 BE 76 64 BD 46 64 E0
62E0- 01 F0 15 AA E0 0A B0 05
62E8- A9 B0 20 5C DB A9 00 20
62F0- 24 ED 68 A8 88 10 CF 60
62F8- 0A 6D 47 64 AA A0 03 BD
6300- 4A 64 20 5C DB E8 88 D0
6308- F6 F0 E7 AD 4C 64 F0 1A
6310- 78 AD 2C 63 8D FE 03 AD
6318- 2D 63 8D FF 03 A0 19 20
6320- 2E 63 A9 08 A0 12 20 2E
6328- 63 58 60 4C 46 63 48 A2
6330- 00 86 FE AE 4C 64 86 FF
6338- B1 FE 85 FE 8A 0A 0A 0A
6340- 0A A8 68 6C FE 00 78 A5
6348- FE 8D 85 64 A5 FF 8D 86
6350- 64 A5 45 8D 87 64 8E 88
6358- 64 8C 89 64 A9 00 A0 12
6360- 20 2E 63 A0 13 20 2E 63
6368- 90 16 AD 85 64 85 FE AD
6370- 86 64 85 FF AC 89 64 AE
6378- 88 64 AD 87 64 4C EB BF
6380- A5 22 C9 01 F0 05 A9 00
6388- 8D 8A 64 CE 8D 64 F0 03
6390- 4C 18 64 AD 8C 64 8D 8D
6398- 64 AD 8A 64 F0 16 A2 0D
63A0- 2C 8B 64 10 07 2C 1F C0
63A8- 10 02 A2 06 A9 1A 5D 00
63B0- 04 9D 00 04 CE 8E 64 D0
63B8- 5F A9 78 8D 8E 64 EE 92
63C0- BF AD 92 BF C9 3C D0 18
63C8- A9 00 8D 92 BF EE 93 BF
63D0- AD 93 BF A0 B9 C9 18 D0
63D8- 07 A9 00 8D 93 BF A0 B3
63E0- AD 8A 64 F0 33 8C 90 64
63E8- A0 0F 98 2C 8B 64 10 0B
63F0- 2C 1F C0 10 06 4A B0 03
63F8- 2C 55 C0 AA FE 00 04 B9
6400- 84 64 DD 00 04 B0 11 A9
6408- B0 9D 00 04 88 C0 0D F0
6410- FB 2C 54 C0 C0 0A D0 D2
6418- 2C 54 C0 A9 08 A0 12 20
6420- 2E 63 AD 85 64 85 FE AD
6428- 86 64 85 FF AC 89 64 AE
6430- 88 64 AD 87 64 58 40 00
6438- CC C9 C2 C5 D2 C5 01 C8
6440- C5 D5 D2 C5 04 00 00 00
6448- 00 00 00 00 00 CA C1 CE
6450- C6 C5 D6 CD C1 D2 C1 D6
6458- D2 CD C1 C9 CA D5 CE CA
6460- D5 CC C1 CF D5 D3 C5 D0
6468- CF C3 D4 CE CF D6 C4 C5
6470- C3 BA A0 AF AF A0 04 03
6478- 00 01 02 63 0C 1F 17 3B
6480- 00 01 01 00 00 00 00 00
6488- 00 00 00 00 00 1E 78 B2
6490- B9 00 B5 B9
```

# Tokenisation de chaînes et équations paramétriques

G Zwingelstein

Ayant apprécié et utilisé le programme *Tokfon*, permettant de modifier la définition d'une fonction en Basic proposé par Gilles Mauffrey dans le numéro 11 de Pom's sous le titre : "Tracé de courbe en conversationnel", j'ai néanmoins eu besoin de le modifier, car le programme sur lequel je travaillais nécessitait la modification de deux fonctions prédéfinies.

La version proposée ici permet donc de redéfinir n'importe quelle fonction définie par la commande *DEF FN YY(X)* dans un programme, quelle que soit sa place, en lui fournissant la nouvelle définition de la fonction sous forme d'une chaîne de caractères, cette chaine pouvant résulter d'une formule.

Après avoir BLOADer le programme "MODIF FN", la fonction définie par *DEF FN YY(Z)=...* peut être modifiée par *CALL ZZZZ,X$ FN YY* où :

- ZZZZ est l'adresse où est chargé "MODIF FN", celui-ci étant relogeable (l'adresse par défaut est $300) ;
- X$ est la chaine contenant la nouvelle formule ;
- YY est le nom de la fonction prédéfinie.

Le programme dont le source "MODIF FN.S" a été développé sur l'assembleur MERLIN se charge alors de "tokeniser" la chaîne, puis de la mettre à la place de l'ancienne définition, à condition que :

- la chaine évaluée n'ait pas une longueur nulle (ILLEGAL QUANTITY ERROR) ;
- la fonction ait déja été définie (UNDEF FUNCTION ERROR) ;
- la place soit suffisante (STRING TOO LONG).

Il termine ensuite si nécessaire la définition par deux points puis le token de REM, pour que le programme BASIC n'interprète pas les codes pouvant résulter d'une définition plus courte que les précédentes.

Comme exemple d'utilisation, le programme PARAMETRIQUE permet de tracer des courbes paramétriques définies par deux fonctions d'un paramètre T, l'une suivant l'axe X horizontal, l'autre suivant l'axe Y vertical.

## Programme 'PARAMETRIQUE'

```
10  GOTO 1000
1000  REM PROGRAMME PRINCIPAL
1010  TEXT : HOME
1020  DEF  FN X(T) = T: REM @@@@@@@@@@@@@@@@@@@@@@@
      @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
      @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
      @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
      @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
      @@@@@@@@@@@@@@@@@@@@@@@
1030  DEF  FN Y(T) = T: REM @@@@@@@@@@@@@@@@@@@@@@@
      @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
      @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
      @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
      @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
      @@@@@@@@@@@@@@@@@@@@@@@
1040 X1 = 0:X2 = 279:Y1 = 0:Y2 = 191: REM LIMITES
      DE L'ECRAN
1050 PA = 280: REM NOMBRE DE PAS POUR TRACER LA FO
      NCTION
1055  REM IL FAUDRA ATTENDRE ENVIRON :
1056  REM PA / 20 +
1057  REM NOMBRE DE FONCTIONS TRANSCENDANTES (COS,
      SIN,EXP...) * PA / 40 +
1058  REM NOMBRE DE DIVISIONS ET MULTIPLICATIONS *
       PA / 400
1059  REM SECONDES
1060  DIM X(PA),Y(PA)
1070 D$ =  CHR$ (4)
1080  PRINT D$"BLOAD MODIF FN"
1100  REM BOUCLE DES FONCTIONS
1110  GOSUB 2000
1120  GOSUB 3000
1130  GOSUB 4000
1140  GOSUB 5000
1150  GOSUB 6000
1160  GET A$
1170  TEXT
1200  REM AUTRE FONCTION OU VALEUR
1210  VTAB 24: HTAB 1: PRINT "D'AUTRES VALEURS POU
      R T (O/N) : ";
1220  GET A$: IF A$ <  > "O" AND A$ <  > "N" THEN
      1220
1230  HTAB 1: CALL  - 958
1240  IF A$ = "O" THEN 1120
1250  VTAB 24: HTAB 1: PRINT "D'AUTRES FONCTIONS (
      O/N) : ";
1260  GET A$: IF A$ <  > "O" AND A$ <  > "N" THEN
      1260
1270  HTAB 1: CALL  - 958
1280  IF A$ = "O" THEN 1110
1990  HOME : END
2000  REM ENTREE DES FONCTIONS
2010  VTAB 1: HTAB 5: PRINT "TRACE D'UNE COURBE PA
      RAMETRIQUE"
2020  VTAB 4: PRINT "DONNER LES EQUATIONS EN FONCT
      ION DU TEMPS (EXEMPLE : '3*COS(T)+T')"
```

```
2030  GOSUB 2100
2040  GOSUB 2200
2050  RETURN
2100  REM
2110  VTAB 8: HTAB 1: PRINT "DONNER LA FONCTION PO
      UR L'AXE DES X :"
2120  VTAB 10: PRINT F1$
2130  VTAB 10: INPUT "";F1$
2140  RETURN
2200  REM
2210  VTAB 13: HTAB 1: PRINT "DONNER LA FONCTION P
      OUR L'AXE DES Y :"
2220  VTAB 15: PRINT F2$
2230  VTAB 15: INPUT "";F2$
2240  RETURN
2250  RETURN
3000  REM ENTREE DES LIMITES POUR T
3010  VTAB 18: HTAB 1: PRINT "VALEUR MIN POUR T :
      ";T1;
3020  HTAB 21: INPUT "";T$
3030 T1 =  VAL (T$)
3040  VTAB 20: PRINT "VALEUR MAX POUR T : ";T2;
3050  HTAB 21: INPUT "";T$
3060 T2 =  VAL (T$)
3070  IF T2 = T1 THEN MG$ = "VALEUR MIN ET MAX DIF
      FERENTES": GOSUB 7000: GOTO 3000
3080  GOSUB 8000: RETURN
4000  REM INSERTION DES FONCTIONS
4010  ONERR  GOTO 4100
4020  CALL 768,F1$ FN X
4030  CALL 768,F2$ FN Y
4040  POKE 216,0
4050  RETURN
4100  REM ERREUR INSERTION FONCTION
4110  CALL 62248: REM REMET LE STACK POINTER COMME
       IL ETAIT AVANT L'ERREUR
4120  GOSUB 10000
4130 A$ = "X": IF EL = 4030 THEN A$ = "Y"
4140  IF ER = 53 THEN MG$ = "PAS DE FONCTION " + A
      $
4150  IF ER = 176 THEN MG$ = "FONCTION " + A$ + "
      TROP LONGUE"
4160  GOSUB 7000
4170  IF A$ = "X" THEN  GOSUB 2100
4180  IF A$ = "Y" THEN  GOSUB 2200
4190  GOSUB 8000: RESUME
5000  REM CALCUL MAX. ET MIN.
5010 MG$ = "UN INSTANT SVP": GOSUB 7000
5020  ONERR  GOTO 5200
5030 M1 =  FN X(T1):M2 = M1
5040 N1 =  FN Y(T1):N2 = N1
5050  FOR I = 0 TO PA
5060 T = T1 + I * (T2 - T1) / PA
5070 X(I) =  FN X(T)
5080 Y(I) =  FN Y(T)
5090  IF X(I) < M1 THEN M1 = X(I)
5100  IF X(I) > M2 THEN M2 = X(I)
5110  IF Y(I) < N1 THEN N1 = Y(I)
5120  IF Y(I) > N2 THEN N2 = Y(I)
5130  NEXT I
5140 A = 1 / (M2 - M1)
5150 A = 1 / (N2 - N1)
5160  GOSUB 8000
5170  POKE 216,0
5180  RETURN
5200  REM ERREUR CALCUL FONCTION
5210  CALL 62248: REM CF LIGNE 4110
5220  GOSUB 10000
5230  IF EL = 5140 THEN M2 = M1 + 1: RESUME
5240  IF EL = 5150 THEN N2 = N2 + 1: RESUME
5250 A$ = "X": IF EL = 5040 OR EL = 5080 OR EL = 5
      150 THEN A$ = "Y"
5260  IF ER <  > 69 AND ER <  > 133 THEN MG$ = "ER
      REUR DE SYNTAX POUR LA FONCTION " + A$: GOTO
       5290
5270  IF ER = 69 THEN MG$ = "DEPASSEMENT POUR LA F
      ONCTION " + A$
5280  IF ER = 133 THEN MG$ = "DIVISION PAR 0 POUR
      LA FONCTION " + A$
5290  GOSUB 7000
5300  IF A$ = "X" THEN  GOSUB 2100
5310  IF A$ = "Y" THEN  GOSUB 2200
5320  POP : GOTO 1130
6000  REM TRACE DE LA COURBE
6010  HGR2 : HCOLOR= 3
6020  HPLOT (X(0) - M1) * (X2 - X1) / (M2 - M1),(Y
      2 - Y1) - (Y(0) - N1) * (Y2 - Y1) / (N2 - N1
      )
6030  FOR I = 1 TO PA
6040  HPLOT  TO (X(I) - M1) * (X2 - X1) / (M2 - M1
      ),(Y2 - Y1) - (Y(I) - N1) * (Y2 - Y1) / (N2
      - N1)
6050  NEXT I
6060  RETURN
7000  REM AFFICHE MESSAGE EN LIGNE 24
7005  GOSUB 8000
7010  INVERSE
7020  VTAB 24: HTAB (40 -  LEN (MG$)) / 2: PRINT M
      G$;
7030  NORMAL : RETURN
8000  REM EFFACE MESSAGE EN LIGNE 24
8010  VTAB 24: HTAB 1: CALL  - 958: RETURN
10000  REM NUMERO ET LIGNE ERREUR
10010 ER =  PEEK (222)
10020 EL =  PEEK (218) + 256 *  PEEK (219)
10030  RETURN
```

# Récapitulation 'MODIF FN'

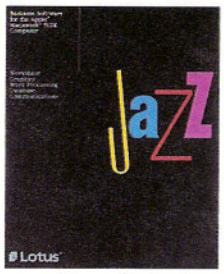
*Après avoir saisi ce code sous moniteur, vous le sauvegarderez par BSAVE MODIF FN,A$300,L$7E*

```
0300- 20 06 E3 20 BE DE 20 7B
0308- DD 20 FD E5 A8 F0 69 A9
0310- 00 99 00 02 88 B1 5E 99
0318- 00 02 88 10 F8 A5 B8 48
0320- A5 B9 48 A9 02 85 B9 A9
0328- 00 85 B8 20 59 D5 84 0F
0330- 68 85 B9 68 85 B8 20 41
0338- E3 A0 01 B1 8A F0 36 85
0340- 5F 88 B1 8A 85 5E B1 5E
0348- F0 03 C8 D0 F9 98 AA F0
0350- 2A 18 69 05 38 E5 0F 90
0358- 22 A0 00 B9 00 02 F0 08
0360- 91 5E C8 CA D0 F5 F0 0C
0368- A9 3A 91 5E C8 CA F0 04
0370- A9 B2 91 5E 60 4C 0E E3
0378- 4C 99 E1 4C B2 E5
```

# Source 'T.MODIF FN'
## Assembleur MERLIN

```
 1 ********************************
 2 *                              *
 3 *    INSERTION D'UNE CHAINE    *
 4 * TOKENISEE DANS LA DEFINITION *
 5 *  D'UNE FONCTION DANS UN PRMG *
 6 *                              *
 7 ********************************
 8
 9 * (PROGRAMME RELOGEABLE)
10
11 * SYNTAX : CALL 768,X$ FN YY
12 * TRANSFORME EN TOKEN LA FORMULE DANS X$, ET
13 * LA DEPLACE DANS LA DEFINITION DE LA FONCTIO
14 * N YY (X$ PEUT ETRE UNE FORMULE DONNANT UNE
15 * CHAINE, EX : CALL 768,A$+"+"+B$ FN YY   )
16
17 REM      =    $B2         ;TOKEN DE REM
18
19 PNTR     =    $0F         ;CONTIENT LG CHAINE
20 INDEX    =    $5E
21 FNCNAM   =    $8A         ;CONTIENT ADR VARIAB
                  LE FONCTION
22 TXTPTR   =    $B8
23
24 IN       =    $200        ;BUFFER ENTREE
25 GETIN    =    $D559       ;ROUTINE DE
26                           ;CONVERSION EN TOKEN
27 PARSE    =    $D56C       ;CONVERTIS EN TOKEN
                  A PARTIR DE $200+X
28 CHKSTR   =    $DD6C       ;CHECK LAST RESULT =
                   STRING
29 FRMEVL   =    $DD7B       ;EVALUE FORMULE
30 CHKCOM   =    $DEBE       ;CHECK ','
31 PTRGET   =    $DFE3       ;RECUPERER ADRESSE V
                  AR. A TXTPTR
32 IQERR    =    $E199       ;"ILLEGAL QUANTITY"
33 ERRDIR   =    $E306       ;CHECK MODE PROGRAMM
34 UNDFNC   =    $E30E       ;"UNDEF FUNCTION"
35 FNC?     =    $E341       ;CHECK TOKEN DE 'FN'
36                           ;ET RETOURNE LE POIN
37                           ;TEUR VERS LA VARIAB
                  LE CONTENANT  LES PARAMETRES D
                  E LA FONCTION
38 STRLONG  =    $E5B2       ;"STRING TOO LONG"
39 FRESTR   =    $E5FD       ;CHECK STRING ET RET
                  OURNE ADRS ET LONG
40
41 * UNE VARIABLE REPRESENTANT UNE FONCTION
42 *  (EX : DEF FN AB(CD)=...) EST CODEE :
43 * - 1IER CAR EN ASCII NEGATIF (= "A")
44 * - 2IEME CAR EN ASCII POSITIF (= 'B')
45 * - ADRESSE FORMULE DE LA FONCTION (2 OCTETS)
46 * (POINTE APRES SIGNE = DANS DEF FN AB(CD)=.)
47 * - ADRESSE ARGUMENT INFORMEL 'CD' (2 OCTETS)
48 * - 1IER CAR DE L'ARGUMENT (='C')
49
50          ORG  $300
51
52          JSR  ERRDIR      ;ERREUR MODE DIRECT
53 * RECUPERE CHAINE
54          JSR  CHKCOM
55          JSR  FRMEVL
56          JSR  FRESTR
57          TAY
58          BEQ  ERROR       ;LONGUEUR = 0
59 * LA TRANSFERT DANS BUFFER IN
60          LDA  #0
61          STA  IN,Y        ;RAJOUTE '00' DE FIN
62          DEY
63 LOOP     LDA  (INDEX),Y
64          STA  IN,Y
65          DEY
66          BPL  LOOP
67 * TOKENISE LA FONCTION
68          LDA  TXTPTR
69          PHA
70          LDA  TXTPTR+1
71          PHA
72          LDA  #>IN
73          STA  TXTPTR+1    ;(FACULTATIF : NON U
                 TILISE DANS GETIN)
74          LDA  #<IN
75          STA  TXTPTR
76          JSR  GETIN       ;Y = NBRE CAR DE FN
77          STY  PNTR        ;SAUVE LONGUEUR FN
78          PLA
79          STA  TXTPTR+1
80          PLA
81          STA  TXTPTR
82 * RECUPERE ADRESSE FONCTION
83          JSR  FNC?
84          LDY  #1
85          LDA  (FNCNAM),Y
86          BEQ  UNDEF       ;FN NON DEFINIE
87          STA  INDEX+1
88          DEY
89          LDA  (FNCNAM),Y
90          STA  INDEX
91 * VERIFIE QUE LA PLACE EST SUFFISANTE
92 CHKPLCE  LDA  (INDEX),Y
93          BEQ  FINLIGNE    ;(0=FIN LIGNE BASIC)
94          INY
95          BNE  CHKPLCE     ;TOUJOURS
96 *
97 FINLIGNE TYA
98          TAX
99          BEQ  ERROR2      ;PAS DE PLACE
100         CLC
101         ADC  #5          ;AJUSTE POUR PNTR
102         SEC
103         SBC  PNTR
104         BCC  ERROR2      ;PAS ASSEZ DE PLACE
105 * TRANSFERT NOUVELLE FONCTION
106         LDY  #0
107 LOOP2   LDA  IN,Y
108         BEQ  FINIR
109         STA  (INDEX),Y
110         INY
111         DEX
112         BNE  LOOP2
113         BEQ  END         ;PAS DE ': REM'
114 * RAJOUTE ': REM' SI NECESSAIRE
115 FINIR   LDA  #':'
116         STA  (INDEX),Y
117         INY
118         DEX
119         BEQ  END
120         LDA  #REM
121         STA  (INDEX),Y
122 *
123 END     RTS
124
125 * TRAITEMENT ERREURS
126 UNDEF   JMP  UNDFNC      ;ERR "UNDEF FUNCTION
127
128 ERROR   JMP  IQERR       ;"ILLEGAL QUANTITY"
129
130 ERROR2  JMP  STRLONG     ;"STRING TOO LONG"
```

# Tant qu'à faire 5 choses à la fois,




Que ceux qui aiment travailler en faisant deux ou trois choses à la fois ne changent rien, au contraire, avec Apple et Jazz ils peuvent faire mieux.

Jazz de Lotus, c'est un programme créé pour Macintosh 512 Ko, équipé d'un lecteur externe qui permet de devenir un parfait jongleur professionnel.

Cinq programmes en concert, c'est-à-dire un tableur, un grapheur, un gestionnaire de fichier, un traitement de texte et un programme de communication réunis en un seul programme. Ou comment être à cinq sur la même souris.

Pouvoir gérer cinq applications à partir d'un seul écran, modifier une donnée dans une application et qu'elle se modifie automatiquement dans les autres, pouvoir sauter d'un graphe à un traitement de texte sans attendre les secondes qui durent une éternité pour changer de programme, c'est bien...

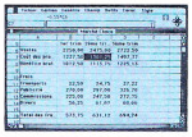
Quand on s'aperçoit que ces cinq programmes sont individuellement excellents, c'est une révolution.

Jazz, c'est l'outil idéal d'un directeur de service.

Prenons un exemple qui exige beaucoup de souplesse, de rapidité, et de doigté, la direction des services secrets :

– Avec un tableur capable d'afficher 8192 lignes sur 256 colonnes, la gestion devient beaucoup plus facile. Si un fait nouveau apparaît, rien n'est plus facile que d'étudier deux ou trois hypothèses afin de ne pas être pris de court.

– Quand un projet est fin prêt, il faut bien le présenter à ses supérieurs et parfois même beaucoup plus haut; un grapheur permet de transformer toutes les données numériques obscures en graphiques lumineux.

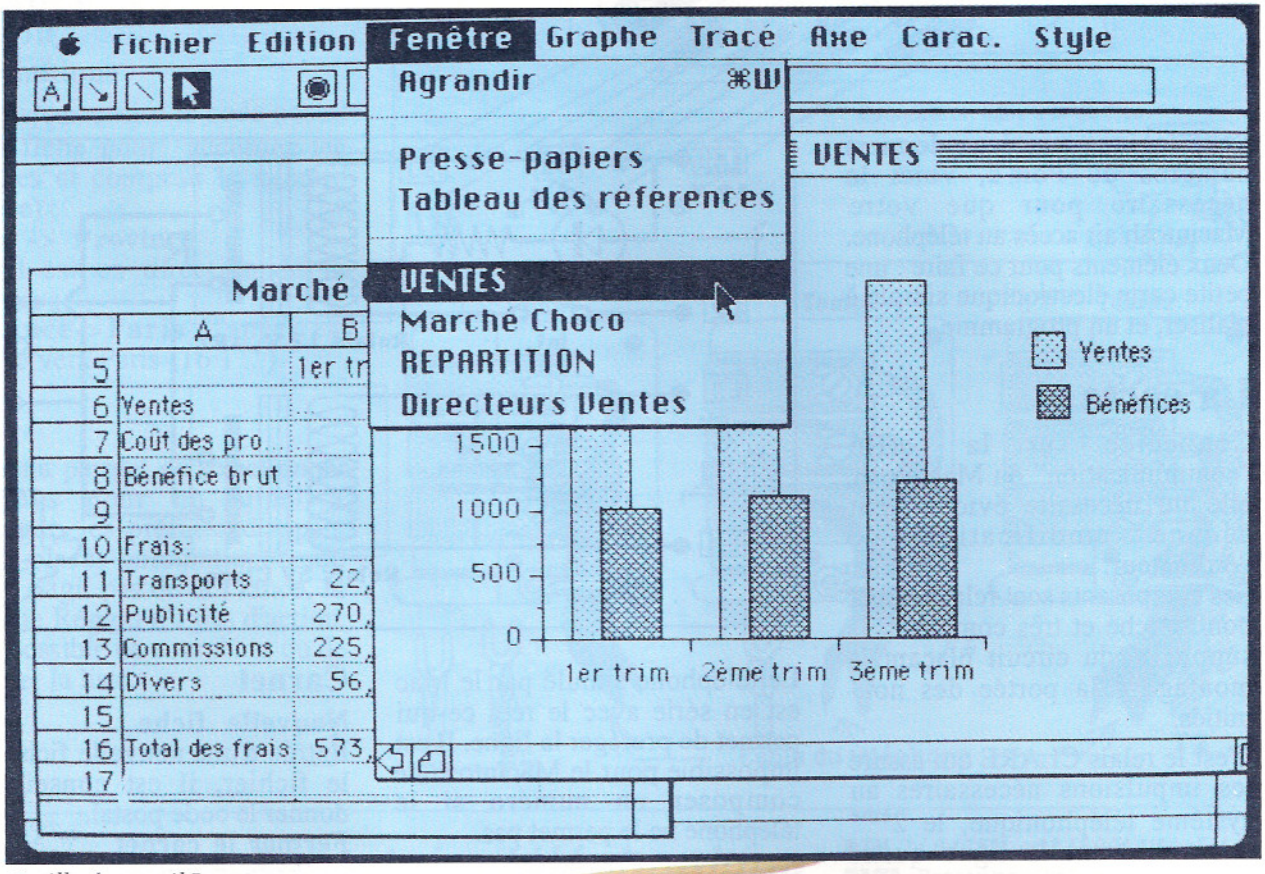


– Si le projet est accepté, un traitement de texte est nécessaire pour que chaque agent concerné soit au courant dans ses moindres détails. Evidemment quand on travaille dans un tel service il vaut mieux pour Macintosh pouvoir copier, effacer, modifier, ou coller des dessins, à chaque instant.

C.L.M. & BBDO. Macintosh est une marque sous licence chez Apple Computer, Inc.

Apple Computer France. Avenue de l'Océanie -

# autant les faire en même temps.




*Feuille de travail Jazz.*

Apple, le logo Apple sont les marques déposées d'Apple Computer, Inc.
Jazz est une marque déposée de Lotus Development Corporation.

Grâce à son gestionnaire de fichier, Jazz fait la fusion automatique entre un mémo et une liste de correspondants à sélectionner selon un critère spécial.
Et avec un modem full duplex 1200 bauds en un instant votre correspondant reçoit toutes les informations précises à l'autre bout du monde.
Avec Jazz l'entreprise est rentable, rondement menée, précise et discrète, inutile d'avoir 35 collaborateurs pour rédiger le projet.

Jazz c'est cinq programmes liés entre eux si parfaitement que l'agent double est enfin enterré, place à l'agent quintuple.

Apple

Z.A. de Courtabœuf, B.P. 131, 91944 Les Ulis Cedex.

# Un Composeur Telephonique sur Macintosh

D. Venot

Suite logique aux deux derniers numéros de Pom's, voici le nécessaire pour que votre Macintosh ait accès au téléphone. Deux éléments pour ce faire : une petite carte électronique simple à réaliser, et un programme.

## La carte

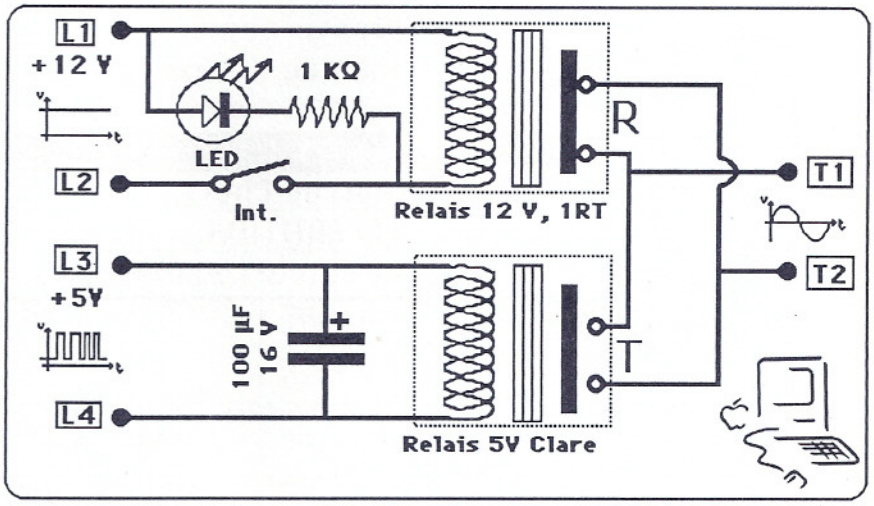
Connectée sur la sortie "communication" du Macintosh, elle ne nécessite évidemment aucune modification de l'ordinateur.
Les composants sont relativement bon marché et très communs, la simplicité du circuit plaçant le montage à la portée des non-initiés.

C'est le relais CLARE qui assure les impulsions nécessaires au système téléphonique, le 2ème relais libèrant la ligne quand l'ordinateur est hors-tension.

Les schémas devraient permettre de mener à bien cette petite réalisation ; le condensateur (anti-parasite) de 100 µF est polarisé donc attention au sens du montage. La sortie destinée à l'ordinateur doit obligatoirement être munie d'une prise 9 broches. L'interrupteur est une sécurité pour couper le système tandis que le Mac utilise un autre programme (incompatible avec le montage.).

**Liste des composants**

- Une résistance 1KΩ
- Un condensateur 100 µF/ 100V
- Une diode LED 5mm
- Un interrupteur
- Un connecteur 9 broches
- Un relais miniature 12V Matsushita HB2
- Un relais 5V CLARE 15005AB




Le téléphone simulé par le Mac est en série avec le réel ce qui permet de protéger la ligne. Il est impossible pour le Macintosh de composer un numéro si le téléphone ne le permet pas.

## Le programme

Il se chargera des préfixes dans le cas des communications : Paris province, province Paris (ajout du 16 ou 16 1) ainsi que dans le cas d'appels à l'étranger.

Le fichier répertoire n'est limité que par la place sur le disque. Les boutons "précédente" et "suivante" permettent de changer de page dans le cas où une ne serait pas suffisante. Pour faire apparaître une page en fonction d'une lettre, cliquez sur l'alphabet la lettre désirée. Les fiches sont classées dans l'ordre alphabétique, l'article de classement est la concaténation "nom/prénom".

## Les menus

### Fichier

**Basic**
retour au basic.
**Finder**
retour au finder.

### Carnet

**Nouvelle fiche**
Entrée d'une nouvelle fiche dans le fichier, il est conseillé de donner le code postal.
**Fermer le carnet**
Le carnet est fermé et il est impossible aux regards indiscrets d'observer les fiches. Pour l'ouvrir, cliquer la souris.

### Edition

**Nouveau répertoire**
Après avoir demandé votre autorisation le présent fichier est détruit au profit d'un nouveau (prudence).

### Click

Lorsqu'une page est affichée, le fait de pointer une fiche et de cliquer permet de la détruire, de la modifier, de composer le numéro... La fonction du click est selectionnée dans le menu Click :
**Afficher**
un click fait afficher la totalité d'une fiche ;
**Editer**
un click édite la fiche pointée en vue de modifications ;
**Détruire**
la fiche pointée est détruite aprés une autorisation ;

**Composer le numéro**
le numéro de la fiche pointée est composé immédiatement avec les conventions imposées par les rubriques "Région et Pays".

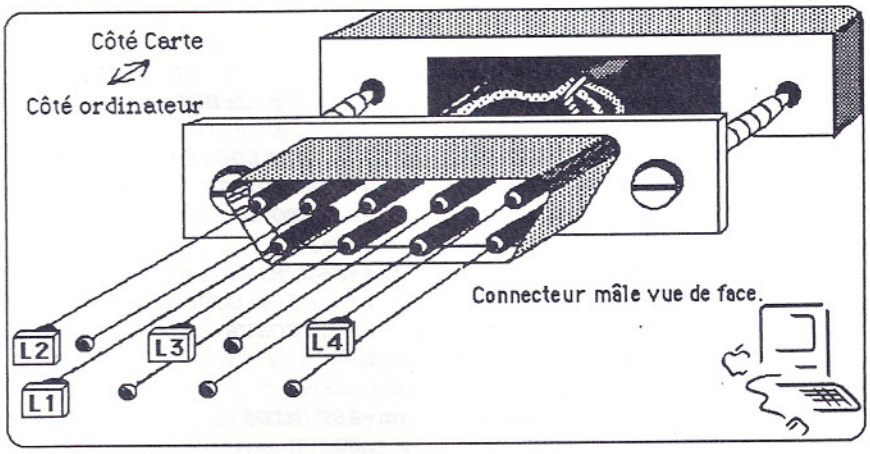
## Région

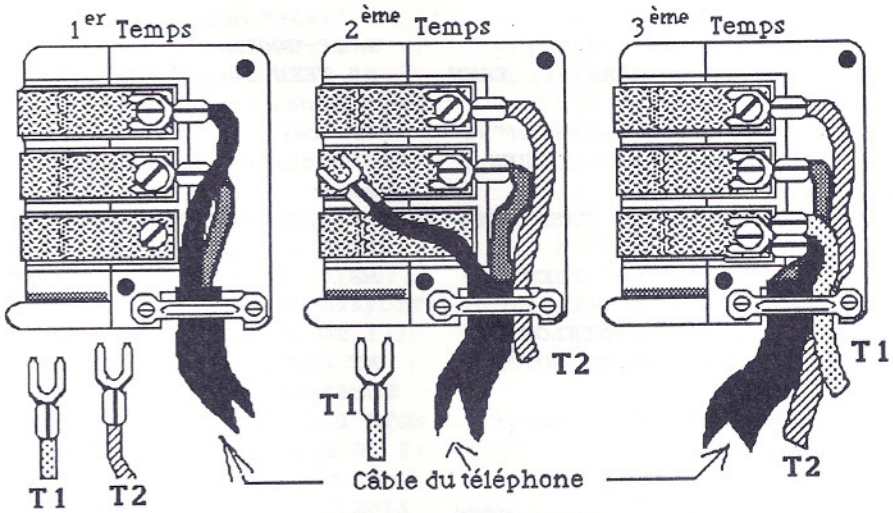
### Automatique

l'ordinateur utilise le code postal de la fiche pour accorder les préfixes et compose le numéro (France) ;
**Paris -> Province**
l'appel est en direction de la province ;
**Province->Paris** L'appel est imposé vers Paris (16 1 ...).

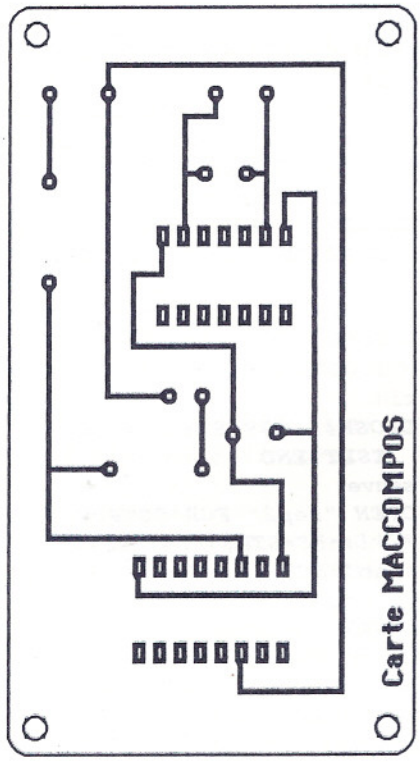
## Pays

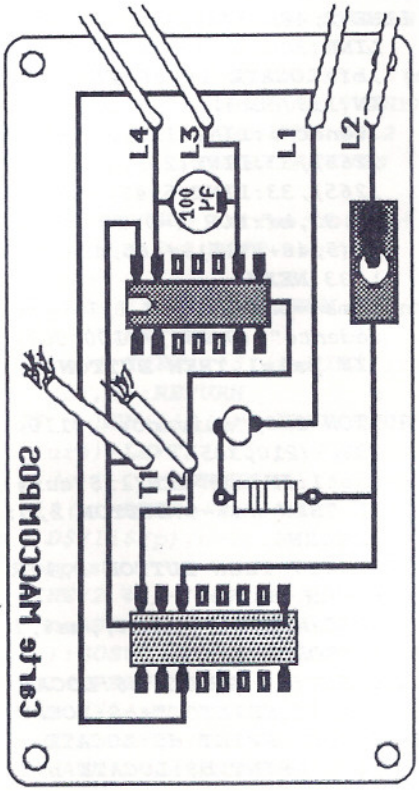
Ce menu permet de sélectionner un pays pour un appel à l'étranger, les préfixes d'appels seront placés automatiquement. S'il ne s'agit pas de la France, la rubrique Région n'a pas d'action. Il est possible de compléter ou de modifier la liste.




## Programme Répertoire téléphonique




```
DIM li$(25),cu%(100):np=5
DATA Royaume Uni,19# 44,Alle
  magne RFA,19# 49,Etats Uni
  s,19# 1,Belgique,19# 32,Su
  isse,19# 41
OPEN "com1:110" FOR OUTPUT
  AS 1:CLOSE
start:
OPEN "rep2" FOR INPUT AS 1
  LEN=113:INPUT#1,nb$:FOR
  i=0 TO 25:LINE INPUT#1,
  li$(i):NEXT:CLOSE#1:OPEN
  "rep1" AS 1:FIELD#1,15 A
  S no$,15 AS pr$,18 AS tel
  $,40 AS adr$,10 AS cp$,15
  AS vil$:nb%=VAL(nb$):cur%
  =0:ra=16:pa%=1
WINDOW 1,"agenda",(8,28)-(5
  05,335),2:WINDOW OUTPUT
  1:TEXTFONT 0:TEXTSIZE 12
  :TEXTMODE 1:GOSUB curseu
  r:FOR i=1 TO 5:MENU i,0,0
  ,"":NEXT:MENU 1,0,1,"Fich
  ier":MENU 1,1,1,"Basic":M
  ENU 1,2,1,"Finder":MENU 2
  ,0,1,"Carnet":MENU 2,1,1,"
  Nouvelle fiche"
MENU 2,2,1,"Fermer le carnet
  ":MENU 3,0,1,"Edition":ME
  NU 3,1,1,"Nouveau répertoi
  re":MENU 4,0,1,"Click":ME
```

```
NU 4,1,1,"Afficher":MENU
4,2,1,"Editer":MENU 4,3,1,
"Détruire":MENU 4,4,2,"Com
poser le numéro":MENU 5,0,
1,"Région"
MENU 5,1,2,"Automatique":ME
NU 5,2,1,"Paris -> Provinc
e":MENU 5,3,1,"Province ->
 Paris":MENU 6,0,1,"Pays":
MENU 6,1,2,"France":RESTO
RE:FOR i=1 TO np:READ a$
,b$:MENU 6,i+1,1,a$:NEXT:
re$="":pa$="":pa=1:re=1:ME
NU ON:etso=3:ON MENU GOS
UB testmenu:GOSUB dessin
IF start=1 THEN RETURN
debut:
MENU:p$=INKEY$:p=ASC(p$+"
")-48:IF p$=CHR$(34) THEN
 p=3
IF p$<>"" THEN a=INSTR("à&
ée'(§è!ç",p$):IF a THEN p
=a-1
IF p>=0 AND p<10 THEN GOSU
B digit
m=MOUSE(0):IF m=1 THEN sou
ris
d1=DIALOG(0):d2=DIALOG(1)
IF d1=1 THEN ON d2 GOTO pr
e,sui
IF d1=5 THEN GOSUB dessin
GOTO debut
pre:
pa%=pa%-6:GOSUB carnet:GOTO
 debut
sui:
pa%=pa%+6:GOSUB carnet:GOTO
 debut
souris:
a=MOUSE(1):b=MOUSE(2):IF b
>48 AND b<260 AND a<460 T
HEN aff
IF b>30 THEN BEEP:GOTO deb
ut
l=cur%:cur%=(a-10)/ra:IF l=c
ur% OR cur%>=26 THEN cur%
=l:BEEP:GOTO debut
PUT (l*ra+4,0),cu%(0),XOR:F
OR i=(l*ra)+4 TO cur%*ra+4
 STEP 3*SGN(cur%-l):PUT (
i,0),cu%(0),XOR:PUT (i,0)
,cu%(0),XOR:NEXT:PUT (cur
%*ra+4,0),cu%(0),XOR:pa%=1
:GOSUB carnet:GOTO debut
aff:
n=(b-64)/32+pa%:IF n>LEN(li
$(cur%)) THEN BEEP:GOTO
 debut
IF etso=0 THEN souris2
IF etso=2 THEN detruire
IF etso=3 THEN composer
nn=ASC(MID$(li$(cur%),n,1))
:GOSUB entree:LINE(2,48)-
(469,269),30,bf:LOCATE 5,
5:PRINT no$;" ";pr$:LOCAT
E 6,2:PRINT adr$:LOCATE
7,2:PRINT vil$;" ";cp$:LO
CATE 9,25:PRINT"tel : ";t
el$:BUTTON CLOSE 1:BUTTO
N CLOSE 2:BUTTON 1,1,"ok
",(0,282)-(100,305)
WHILE DIALOG(0)<>1:WEND:B
UTTON CLOSE 1:GOSUB carn
et:GOTO debut
souris2:
fi%=ASC(MID$(li$(cur%),n,1)
):nn=fi%:GOSUB entree:mod
if=1:GOSUB nouvin:GOTO de
but
detruire:
nn=ASC(MID$(li$(cur%),n,1))
:GOSUB entree:t$="La fich
e "+a$+" doit être détruit
e ?":GOSUB accord:IF r$="
n" THEN detruire1
GOSUB del
detruire1:
GOSUB carnet:GOTO debut
dessin:
LINE(3,3)-(443,23),33,bf:LI
NE(0,0)-(440,20),30,bf:LI
NE(0,20)-(440,20),33:LINE
(440,0)-(440,20),33:FOR i=
1 TO 26:LOCATE 1,i*2:PRI
NT CHR$(96+i):NEXT:PUT(c
ur%*ra+4,0),cu%(0),XOR:FO
R i=1 TO 10 STEP 2:LINE(
10-i,40-i)-(480-i,280-i),3
0,bf
LINE(10-i,40-i)-(480-i,280-i
),33,b:NEXT:LOCATE 3,3:P
RINT"Nom :":LOCATE 3,15:
PRINT" Téléphone:":LOCAT
E 3,27:PRINT"Adresse :"
carnet:
LINE(2,48)-(469,269),30,bf:
LINE(300,282)-(500,305),30
,bf:LOCATE 19,40:PRINT L
EN(li$(cur%));"/";nb%;"© D
.Venot";:LINE(115,35)-(115
,265),33:LINE(205,35)-(205
,265),33:LINE(5,48)-(465,4
7),33,bf:FOR i=0 TO 13:LI
NE(5,48+i*16)-(465,48+i*16
),33:NEXT
n=4:ma%=pa%:BUTTON 1,1,"pré
cédente",(0,282)-(100,305)
:IF pa%=1 THEN BUTTON 1,
0
BUTTON 2,0,"suivante",(110,
282)-(210,305)
carnet1:IF ma%>LEN(li$(cur%
)) THEN ma%=0:BUTTON 2,0:
RETURN
IF n>=17 THEN BUTTON 2,1:R
ETURN
nn=ASC(MID$(li$(cur%),ma%,1
)):GOSUB entree
LOCATE n,1:PRINT a$:LOCAT
E n,15:PRINT" "+c$:LOCAT
E n,27:PRINT d$:LOCATE n
+1,3:PRINT b$:LOCATE n+1
,27:PRINT e$+"  "+f$:n=n+
2:ma%=ma%+1:GOTO carnet1
Fermer:
LINE(2,48)-(469,269),30,bf:
FOR i=20 TO 460 STEP 20:
LINE(i,50)-(i,260),33:NEX
T:FOR i=60 TO 260 STEP 20
:LINE(10,i)-(460,i):NEXT:
LINE(30,96)-(200,144),30,b
f:LOCATE 8,10:PRINT"Répe
rtoire":WHILE MOUSE(0)<>1
:WEND:GOTO carnet
curseur:
LINE(0,0)-(20,18),33,b:LINE
(1,1)-(19,17),33,b:LINE(10
,0)-(10,3),33,b:LINE(10,18
)-(10,15),33,b:GET(0,0)-(2
0,18),cu%(0):RETURN
testmenu:
m1=MENU(0):m2=MENU(1):IF m1
=1 THEN ON m2 GOTO fin,fi
nder
IF m1=2 THEN ON m2 GOTO no
uv,fermer
IF m1=3 THEN rep
IF m1=4 THEN clic
IF m1=5 THEN region
IF m1=6 THEN pays
RETURN
region:
MENU 5,re,1:MENU 5,m2,2:re=
m2
IF m2=1 THEN re$=""
IF m2=2 THEN re$="16 "
IF m2=3 THEN re$="16 1 "
RETURN
pays:
MENU 6,pa,1:pa=m2:MENU 6,pa
,2:IF pa=1 THEN pa$="":RE
TURN
RESTORE:FOR i=1 TO pa-1:RE
AD pays$,pa$:NEXT:RETURN
rep:
t$="Création d'un répertoire
 et destruction du résidan
t.":GOSUB accord:IF r$="n
" THEN carnet
CLOSE#1:OPEN "rep2" FOR OU
TPUT AS 1:PRINT#1,"0":nb
%=0:FOR i=0 TO 25:li$(i)=
"":PRINT#1,li$(i):NEXT:C
LOSE#1:KILL "rep1":start=
1:GOTO start
fin:
CLOSE#1:GOSUB sauve:MENU R
ESET:END
sauve:
OPEN "rep2" FOR OUTPUT AS
#1:nb$=STR$(nb%):PRINT#1,
nb$:FOR i=0 TO 25:PRINT#1
,li$(i):NEXT:CLOSE#1:RET
URN
nouv:
modif=0:a$="":b$="":c$="":d$
="":e$="":f$=""
nouvin:
```

```
LINE(2,48)-(469,269),30,bf:
  LOCATE 7,2:PRINT"Prénom
  :":EDIT FIELD 2,b$,(80,96
  )-(280,112):LOCATE 9,2:PR
  INT"Téléphone :":EDIT FIE
  LD 3,c$,(110,128)-(310,144
  ):LOCATE 11,2:PRINT"Adre
  sse :":EDIT FIELD 4,d$,(8
  0,160)-(440,176):LOCATE 1
  3,2:PRINT"Ville :
EDIT FIELD 5,e$,(70,192)-(2
  70,208):LOCATE 15,2:PRIN
  T"Code postal :":EDIT FIE
  LD 6,f$,(120,224)-(320,240
  ):LOCATE 5,2:PRINT"Nom :
  ":EDIT FIELD 1,a$,(60,64)
  -(260,80):BUTTON 1,1,"Ok"
  ,(0,282)-(50,305):BUTTON
  2,1,"Fin",(60,282)-(130,30
  5)
nouv1:
IF DIALOG(0)<>1 THEN nouv1
bou=DIALOG(1)
IF bou=1 THEN ok
IF bou=2 THEN GOSUB nouvfi
  n:GOTO carnet
GOTO nouv1
ok:
a$=EDIT$(1):b$=EDIT$(2):c$=
  EDIT$(3):d$=EDIT$(4):e$=
  EDIT$(5):f$=EDIT$(6):GOS
  UB nouvfin:fl%=fi%:nb%=nb%
  +1:fi%=nb%
nouvdel:
a$=a$+"  ":a$=UCASE$(LEFT$
  (a$,1))+RIGHT$(a$,LEN(a$)
  -1):n=ASC(a$)-65:IF fi%=1
  3 THEN fi%=0
IF li$(n)="" THEN li$(n)=CH
  R$(fi%):nn=fi%:GOTO sorti
  e
FOR i=1 TO LEN(li$(n)):nn=A
  SC(MID$(li$(n),i,1)):IF n
  n=0 THEN nn=13
GET#1,nn:IF UCASE$(no$+pr$
  )>UCASE$(a$+b$) THEN plac
  e
NEXT
li$(n)=li$(n)+CHR$(fi%):nn=f
  i%:GOTO sortie
place:
nn=fi%:IF i=1 THEN li$(n)=C
  HR$(fi%)+li$(n):GOTO sort
  ie
li$(n)=LEFT$(li$(n),i-1)+CH
  R$(fi%)+MID$(li$(n),i)
sortie:
IF nn=0 THEN nn=13
LSET no$=a$:LSET pr$=b$:LS
  ET tel$=c$:LSET adr$=d$:L
  SET vil$=e$:LSET cp$=f$:P
  UT#1,nn:IF modif=1 THEN n
  n=fl%:GOSUB del:GOTO carn
  et
IF modif=2 THEN RETURN
IF modif=0 THEN nouv
GOTO carnet
nouvfin:
FOR i=1 TO 6:EDIT FIELD CL
  OSE i:NEXT:BUTTON CLOSE
   1:BUTTON CLOSE 2:RETUR
  N
entree:
IF nn=0 THEN nn=13
GET#1,nn:a$=no$:b$=pr$:c$=te
  l$:d$=adr$:e$=vil$:f$=cp$:
  RETURN
clic:
MENU 4,1,1:MENU 4,2,1:MENU
   4,3,1:MENU 4,4,1:IF m2=2
   THEN etso=0:MENU 4,2,2
IF m2=1 THEN etso=1:MENU 4,
  1,2
IF m2=3 THEN etso=2:MENU 4,
  3,2
IF m2=4 THEN etso=3:MENU 4,
  4,2
RETURN
finder:
CLOSE#1:GOSUB sauve:SYSTEM
accord:
LINE(2,48)-(469,269),30,bf:
  LOCATE 5,2:PRINT t$:BUTT
  ON 1,1,"Oui",(100,100)-(15
  0,120):BUTTON 2,1,"Non",(
  170,100)-(220,120):WHILE
  DIALOG(0)<>1:WEND:BUTTON
   CLOSE 1:BUTTON CLOSE 2:
  IF DIALOG(1)=1 THEN r$="
  o":RETURN
r$="n":RETURN
DEL:
IF nn=0 THEN nn=13
fi%=nn:n=INSTR(li$(cur%),CH
  R$(nn)):p=cur%:GOSUB coup
  er:IF nn=nb% THEN nb%=nb%
  -1:RETURN
FOR p=0 TO 25:n=INSTR(li$(p
  ),CHR$(nb%)):IF n<>0 THEN
   GOSUB couper:GOTO del2
NEXT p:BEEP:BEEP:RETURN
del2:
nn=nb%:GOSUB entree:nb%=nb%
  -1:modif=2:GOTO nouvdel
couper:
IF LEN(li$(p))=1 THEN li$(p
  )="":RETURN
IF n=1 THEN li$(p)=MID$(li$
  (p),2):RETURN
IF n=LEN(li$(p)) THEN li$(c
  ur%)=LEFT$(li$(p),LEN(li$
  (p))-1):RETURN
li$(p)=LEFT$(li$(p),n-1)+MI
  D$(li$(p),n+1):RETURN
composer:
LINE(2,48)-(469,269),30,bf:n
  n=ASC(MID$(li$(cur%),n,1)
  ):GOSUB entree:LOCATE 14
  ,4:PRINT A$;" ";B$:LOCAT
  E 15,4:PRINT d$:p$=c$:a$=
  " "+LEFT$(f$,2):B$=RE$:LO
  CATE 16,4:PRINT e$;"
  ";f$:IF pa<>1 THEN b$=""
IF pa=1 AND re=1 THEN IF I
  NSTR(" 91 92 93 95 77 78 9
  4 75   ",a$)<>0 THEN b$=""
   ELSE b$="16 "
LOCATE 5,20:IF pa=1 THEN P
  RINT"En france" ELSE PRI
  NT pays$:GOTO comp1
LOCATE 5,2:IF re=1 THEN PR
  INT"Automatique."
LOCATE 6,7:IF b$="16 " THE
  N PRINT"Vers la province.
  "
IF b$="16 1 " THEN PRINT"V
  ers la région Parisienne."
IF b$="" THEN PRINT"zone i
  ntérieure, région Parisien
  ne."
comp1:p$=pa$+" "+b$+"   "+p$
  :LOCATE 10,3:PRINT"Compo
  ser le numéro :";p$:LOCAT
  E 12,12:PRINT"C O M P O S
   I T I O N":GOSUB numero:
  IF p$<>"" THEN LOCATE 13
  ,10:PRINT"E N    L I G N
  E . . . " ELSE LOCATE 13,
  10:PRINT"I N T E R R U P
  T I O N":GOTO comp2
FOR i=1 TO 5:BEEP:NEXT:FOR
   i= 1 TO 5000:NEXT
comp2:
GOSUB carnet:GOTO debut
numero:
FOR X=1 TO LEN(p$):p=ASC(M
  ID$(p$,x,1))-48:IF p<0 OR
   p>9 THEN impnu
GOSUB digit
IF MOUSE(0)=2 OR INKEY$="
  " THEN p$="":RETURN
num1:NEXT:RETURN
impnu:
IF p=-13 THEN FOR i=1 TO 1
  0000!:NEXT
GOTO num1
DIGIT:
IF p=0 THEN p=10
a$=STRING$(p,248):OPEN "CO
  M1:110,n,8 ,1 " FOR OUTPU
  T AS 2:PRINT#2,A$;:CLOSE
  #2:FOR i=1 TO 1000:NEXT:R
  ETURN
```

## Programme 'Initialise'

```
REM Programme créateur de fi
  chiers "répertoires", doit
   être utilisé avant la pre
  mière mise en service du p
  rogramme
OPEN "rep2" FOR OUTPUT AS
  1:PRINT#1,"0":FOR i=0 TO
  25:PRINT#1,"":NEXT:CLOSE
  :END
```

# Le Macintosh et le Presse-Papiers (fable)

*Jean-Luc Bazanegue*

Le Macintosh offre à ses utilisateurs la remarquable possibilité de faire passer à peu près tout entre applications et/ou accessoires de bureau, et en particulier du texte. Pour cela, les programmeurs qui 'développent' en Basic ou en Pascal (ou avec un autre langage évolué) disposent de routines qu'il suffit d'appeler aux moments voulus, moments qui, nous le verrons plus loin, devront être choisis avec soin si l'on veut respecter la philosophie Macintosh et, surtout, le confort de l'utilisateur. Nous ne nous intéresserons ici qu'aux données de type "TEXT" bien qu'il soit bien entendu possible de passer des informations graphiques "PICT" (picture) par ce biais, mais, de toute façon et pour ce genre de traitement, la principale différence entre les deux types est que c'est exactement la même chose.

## TEScrap et DeskScrap

La plupart des applications et accessoires de bureau utilisent un "presse-papiers" géré exclusivement et automatiquement par le système d'édition de texte (TextEdit). Celui-ci, baptisé **TEScrap** se charge de la manipulation des textes coupés, copiés ou collés à l'aide des routines _TECut, _TECopy et _TEPaste. Donc, à l'intérieur d'une application ou d'un accessoire, le programmeur doit seulement appeler les trois routines en question, en fonction de l'article sélectionné par l'utilisateur dans le menu "Édition" (ou de leur 'équivalent clavier' : Commande-X, C ou V) sans se soucier de la manipulation du texte traité. Mais ceci ne vaut qu'à l'intérieur d'une même application ou d'un même accessoire. Dès lors qu'il s'agit de passer du texte hors du programme en cours, il faut avoir recours à un second presse-papiers : celui du bureau, baptisé **DeskScrap**. Ce dernier est donc employé comme tampon (en mémoire, ou sur disque si sa taille ne permet pas un maintien en mémoire) lorsque l'utilisateur a, par exemple, copié l'heure et la date affichées par l'accessoire "Horloge" pour le coller dans un document MacDraw, ou encore couper un paragraphe de texte depuis MacWrite pour le coller dans le "Calepin" ou dans un document MacPaint.

## Des routines...

...dites de haut niveau pour le Basic, Pascal et les autres langages évolués, mais pas pour l'assembleur, d'où la présence de cet article.

En Basic Microsoft, DeskScrap est considéré comme étant un fichier séquentiel, au même titre que le clavier, les connecteurs 'série' et l'écran. Pour ouvrir ce fichier en entrée :

```
OPEN "CLIP:TEXT" FOR
OUTPUT AS 1
```

en sortie :

```
OPEN "CLIP:TEXT" FOR
INPUT AS 1
```

Les Pascaliens, de leur côté, disposent des routines **TEFromScrap** - pour faire passer du texte depuis le presse-papiers *DeskScrap* vers le presse-papiers *TEScrap* - et **TEToScrap** - pour faire l'inverse.

Rien de tel donc pour l'assembleur et, si 'Inside Macintosh' préconise l'utilisation des variables globales *TEScrpHandle* et *TEScrpLength* ainsi que des fonctions *GetScrap* et *PutScrap*, il n'en reste pas moins que l'écriture de routines équivalentes à TEFromScrap et TEToScrap, bien que relativement simple, n'est pas forcément évidente. C'est pourquoi nous proposons deux routines en assembleurs qui pourront être très facilement implantées dans vos programmes et accessoires de bureau.

## Routine TEFromScrap

Cette routine se charge du passage du texte depuis DeskScrap vers TEScrap. Elle sera appelée depuis votre application ou accessoire par un

```
BSR    TEFromScrap
```

Les registres dont le contenu est susceptibles d'être modifié sont préservés et restitués en fin d'exécution. Au retour, le bit Z du registre d'état est positionné à 1 s'il n'y avait pas de texte à transféré, ou si le texte n'avait pu être transférer du fait d'une erreur au niveau du système d'exploitation (OSErr). En cas d'erreur de ce type, votre programme peut trouver le code d'erreur en -26(SP) à condition, bien sûr, de le faire avant toute nouvelle opération pouvant engendrer une modification du contenu du registre A7 (pointeur de pile).

Voici un exemple de segment de code extrait d'un programme faisant appel à la routine TEFromScrap :

```
        BSR    TEFromScrap
        BEQ.S  Erreur
Suite   ...    ...
        ...    ...
        ...    ...
Erreur  TST    -26(SP)
        BEQ.S  Suite
```

```
        MOVE    -26(SP),AdrEr
        ...     ...
        ...     ...
```

## Routine TEToScrap

TEToScrap, vous l'aurez deviné, effectue l'inverse de la tâche assurée par TEFromScrap, soit le passage du presse-papiers TEScrap vers le presse-papiers DeskScrap.
L'appel de la routine se fait par :

```
BSR     TEToScrap
```

Les registres sont préservés et leur contenu restitué. Au retour, contrairement à la routine TEFromSrap, le bit Z du registre d'état est positionné à 1 s'il n'y a pas eu de problème. Sinon, le bit Z est à 0 et votre application et/ou accessoire peut trouver, tout comme pour la routine précédente, un éventuel code d'erreur OSErr dans -26(SP)

Exemple de segment de code pour la routine TEToScrap :

```
        BSR     TEFromScrap
        BNE.S   Erreur
Suite   ...     ...
        ...     ...
        ...     ...
Erreur  MOVE    -26(SP),AdrEr
        ...     ...
        ...     ...
```

## Quand utiliser les routines ?

La programmation du Macintosh impose aux 'développeurs' un certain nombre de règles précises décrites dans 'Inside Macintosh' au chapitre 'Macintosh User Interface Guidelines'. Ces conventions, lorsqu'elles sont respectées, autorisent la mise au point de programmes que l'utilisateur final pourra utiliser de façon quasi-instinctive. Par exemple, en tant qu'utilisateurs, nous trouvons qu'il n'y a rien de plus désagréable que d'utiliser un programme qui n'emploie pas les 'équivalents claviers' Commande-X, C et V pour 'Couper', 'Copier' et 'Coller'. Nous avons donc tout intérêt à respecter les conventions Macintosh, même si elles nous paraissent lourdes, voire même contraignantes.

### Démarrage de l'application

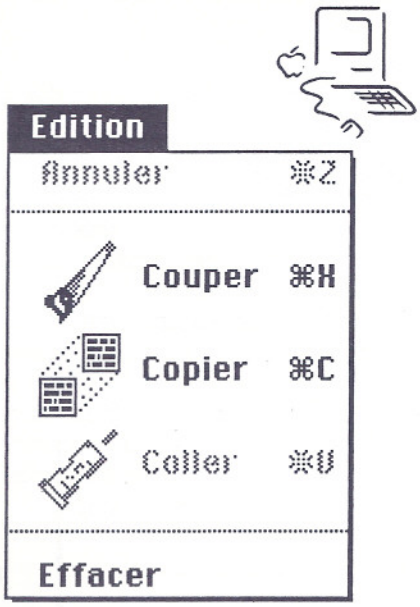
Il serait tentant de transférer immédiatement le contenu de DeskScrap dans TEScrap mais, si nous voulons bien faire les choses, c'est un peu plus compliqué. Il convient tout d'abord de s'assurer de la présence de texte dans DeskScrap. Pour cela, on peut utiliser la routine _GetScrap en passant 0 dans la pile à la place du Handle :

```
SUBQ    #4,SP
CLR.L   -(SP)
MOVE.L  #'TEXT',-(SP)
PEA     scratch8
_GetScrap
MOVE.L  (SP)+,D0
```

En passant une valeur nulle (NIL pour les Pascaliens) dans la pile (CLR -(SP)) à la place d'un Handle sur le presse-papier TEScrap, la routine _GetScrap retourne :

1. -102 si il n'y a pas de donnée de type 'TEXT' (notons bien qu'il ne s'agit pas là d'un code d'erreur : DeskScrap peut très bien contenir à ce moment des données de type 'PICT') ;
2. une autre valeur négative correspondant à une erreur de type OSErr en cas de problème au niveau du système d'exploitation ;
3. la longueur du presse-papiers si la valeur est positive.

Dans les deux premiers cas, il convient de rendre invalide l'article 'Coller' du menu 'Édition' pour que l'utilisateur ne puisse pas croire qu'il peut coller quelque chose alors qu'il n'y a rien dans TEScrap. On pourrait donc obtenir à ce moment un menu 'Édition' comme celui-ci :

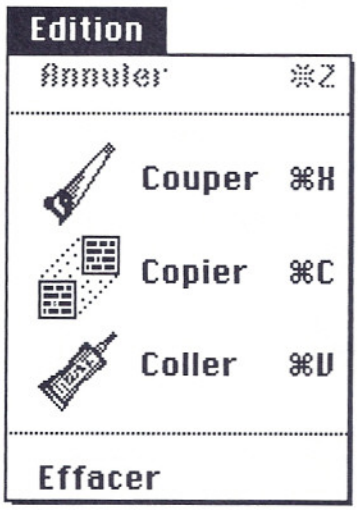



En fait, si l'on veut vraiment bien faire les choses, les menus 'Couper' et 'Copier' ne devraient être valides que s'il y a quelque chose prêt à être copier ou couper ou, en d'autres termes : si quelque chose est sélectionné. Ceci est rendu possible par la présence des champs *selStart* (début de la zone sélectionnée) et *selEnd* (fin de la zone sélectionnée) situés dans l'enregistrement du texte (TERec).

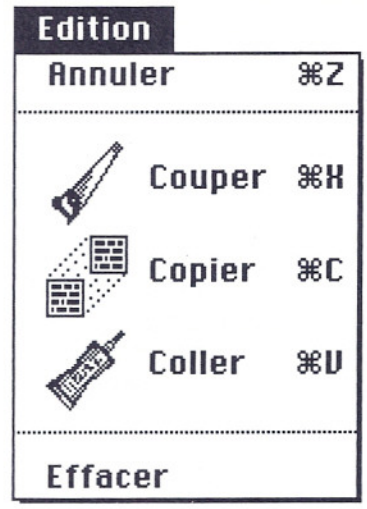
```
MOVEA.L teHandle(A5),A4
MOVEA.L (A4),A4
CMP     selStart(A4),
           selEnd(A4)
BNE     TexteSelectionne
```

En ce qui concerne le second cas, il est recommandé d'afficher une fenêtre de type 'alerte' afin d'informer l'utilisateur en lui indiquant, en fonction du code OSErr retourné, quel est le problème rencontré.

Pour le troisième cas, il convient de valider l'article 'Coller' du menu 'Édition', ce qui pourrait donner un menu comme celui-ci (il n'est pas indispensable de faire des menus aussi délirants !) :




Il n'est pas pour autant indispensable d'appeler la routine TEFromScrap immédiatement. En effet, si l'utilisateur coupe ou copie du texte dans l'application avant le premier 'Coller', le transfert d'informations aura été inutile. De plus, il convient de garder intact le contenu de DeskScrap pour le cas ou il n'y aurait aucun 'Couper' ou 'Coller' entre le démarrage de l'application et l'ouverture d'un accessoire de bureau, ou encore le démarrage de l'application et sa fermeture.
En résumé : après le démarrage d'une application, il faut appeler TEFromScrap seulement lors du premier 'Coller', et seulement si il n'y a pas eu de 'Couper' ou 'Coller' auparavant (chinois).




## Ouverture ou sélection d'un accessoire

Il faut avant tout faire une opération qui n'a pas de rapport direct avec le contenu des presse-papiers : la validation des menu 'Annuler', 'Couper', 'Copier', 'Coller' et 'Effacer' afin d'autoriser leur utilisation éventuelle par l'accessoire :

Deux cas sont ici envisageables :

1 Il n'y a pas eu de 'Couper' ou 'Copier' depuis le démarrage de l'application ou le dernier retour depuis un accessoire de bureau ;
2 Il y a eu au moins un 'Couper' ou 'Copier' depuis le démarrage de l'application ou le dernier retour depuis un accessoire de bureau.

## *Routine TEFromScrap* *('DeskScrap' vers 'TEdit')*

```
; Trap et adresses utilisées :
.Trap          _GetScrap      $A9FD
scratch8       EQU            $9FA
TeScrpLength   EQU            $AB0
teScrpHandle   EQU            $AB4
```

Dans un fichier "ressource" destiné à une véritable application, nous trouverions ici les ressources de type WIND, MENU, DITL, STR, DLOG, etc.

Clipboard File

```
TEFromScrap
      MOVEM.L  A0-A1/D0-D2,-(SP)   ; Sauvegarde les registres au sommet de la pile.
      SUBQ     #4,SP               ; Réserve 4 octets au sommet de la pile pour la
                                   ; valeur retournée par la routine _GetScrap.
      MOVE.L   teScrpHandle,-(SP)  ; Empile le 'Handle' du presse-papier des
                                   ; routines d'édition de texte (TEdit).
      MOVE.L   #'TEXT',-(SP)       ; Empile le type de donnée requise (texte).
      PEA      scratch8            ; Empile l'adresse du tampon de 8 octets.
      _GetScrap                ; _GetScrap retourne une valeur :
                               ; positive = longueur du presse-papier ;
                               ; -102 = pas de données du type demandé ;
      MOVE.L   (SP)+,D0        ; autre négative = OSErr (erreur syst. exploitation).
      BPL.S    FinFrom         ; Si OK, on conserve la valeur placée dans D0.
      MOVEQ    #0,D0           ; Sinon on indique une longueur nulle.
FinFrom                        ; La variable globale TeScrpLength, qui indique
      MOVE     D0,TeScrpLength  ; la longueur du presse-papier des routines
                                ; d'édition, est mise à jour.
      MOVEM.L  (SP)+,A0-A1/D0-D2 ; Valeurs initiales dans les registres.
      RTS                       ; Retour au programme appelant avec le bit Z à 1
                                ; si il n'y a rien dans le presse-papier.
```

Quel que soit le cas concerné, il faut stocker à une adresse quelconque (mais connue !), la valeur courante de *scrapCount*. Ceci nous permettra de savoir ultérieurement si le contenu de DeskScrap a été modifié par l'accessoire. Pour obtenir la valeur de ce compteur, on peut soit lire le contenu de la variable globale scrapCount ($968) :

```
MOVE    scrapCount,Compt(A5)
```

soit lire la zone scrapCount de l'enregistrement *ScrapStuff* dont l'adresse de base est retournée par la routine _InfoScrap :

```
SUBQ    #4,SP
_InfoScrap
MOVEA.L (SP)+,A0
MOVE    scrapCount(A0),Compt(A5)
; scrapCount=8
```

Pour le premier cas (1), on se contentera de ne rien faire (mis à part le stockage du contenu du compteur).

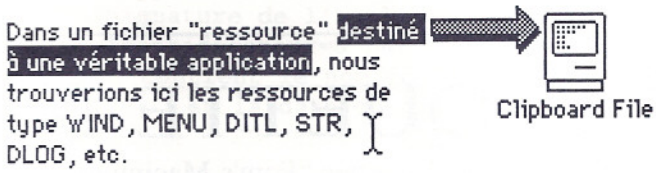
Pour le cas (2), il faudra faire appel à la routine TEToScrap afin de passer le texte copier ou couper à l'accessoire.

## Passage d'un accessoire à l'application

Le problème (si tant est que problème il y a) est le même que lors du démarrage de l'application. Nous disposons toutefois d'un moyen simple pour savoir si le contenu de DeskScrap a été modifié depuis que l'on a quitté l'application : nous avons, lors de l'ouverture ou de la sélection de l'accessoire, stocké

## Routine TEToScrap ('TEdit' vers 'DeskScrap')

```
; Trap et adresses utilisées :
.Trap           _GetHandleSize $A025
.Trap           _HLock         $A029
.Trap           _HUnLock       $A02A
.Trap           _PutScrap      $A9FE
.Trap           _ZeroScrap     $A9FC
TeScrpLength    EQU            $AB0
teScrpHandle    EQU            $AB4
```




```
TEToScrap
      MOVEM.L  A0-A1/D0-D2,-(SP)   ; Sauvegarde les registres au sommet de la pile
      SUBQ     #4,SP               ; Réserve 4 octets au sommet de la pile pour la
                                   ; valeur retournée par la routine _ZeroScrap.
      _ZeroScrap                   ; Initialise le presse-papier du bureau et
                                   ; incrémente ScrapCount de 1 (voir texte).
      ADDQ     #4,SP               ; On n'utilise pas la valeur retournée.
      MOVE.L   teScrpHandle,A0     ; Place le 'Handle' du presse-papier des
                                   ; routines d'édition de texte dans A0.
      SUBQ     #4,SP               ; Réserve 4 octets au sommet de la pile pour la
                                   ; valeur retournée par la routine _PutScrap.
      _GetHandleSize               ; Retourne la longueur du presse-papier des
                                   ; routine d'édition de texte dans D0.
      MOVE.L   D0,-(SP)            ; Empile la longueur.
      _HLock                       ; Empêche le déplacement du presse-papier.
      MOVE.L   #'TEXT',-(SP)       ; Empile le type de donnée à passer.
      MOVE.L   (A0),-(SP)          ; Empile le pointeur sur le texte à passer.
      _PutScrap                    ; Transfert du texte.
      MOVE.L   teScrpHandle,A0     ; Place le 'Handle' du presse-papier des
                                   ; routines d'édition de texte dans A0, et
      _HUnLock                     ; _HUnLock autorise les mouvements de mémoire.
      MOVE.L   (SP)+,D0            ; Résultat de _PutScrap dans D0 pour
                                   ; positionner le bit Z : à 1 si OK.
      MOVEM.L  (SP)+,A0-A1/D0-D2   ; Valeurs initiales dans les registres.
      RTS                          ; Retour au programme appelant avec le bit Z à
                                   ; si l'opération s'est bien déroulée.

; En cas d'erreur au niveau du système d'exploitation (OSErr) et pour les deux
; routines, le code de l'erreur se trouve, lorsque l'on est revenu au programme
; appelant, en -26(SP) sur deux octets.
```

la valeur courante de scrapCount. Or, sans trop entrer dans les détails, on peut dire que ce compteur est incrémenté de 1 à chaque fois que l'on met quelque chose dans DeskScrap. Donc, si le contenu actuel de scrapCount est différent de son contenu avant appel de l'accessoire, le contenu de DeskScrap a été modifié, ce qui doit bien sûr être pris en compte par notre application.

```
SUBQ     #4,SP
_InfoScrap
MOVEA.L  (SP)+,A0
CMP      scrapCount(A0),Compt(A5)
BNE      Modif
BRA      PasModif
```

## Fermeture de l'application

Le traitement doit être logiquement le même que dans le cas "Ouverture ou sélection d'un accessoire", bien qu'il ne soit pas nécessaire de se préoccuper de l'état du menu 'Édition' ni du stockage de la valeur courante de scrapCount qui, de toute manière, ne sera pas réutilisée.

## En cas d'écriture d'un accessoire de bureau

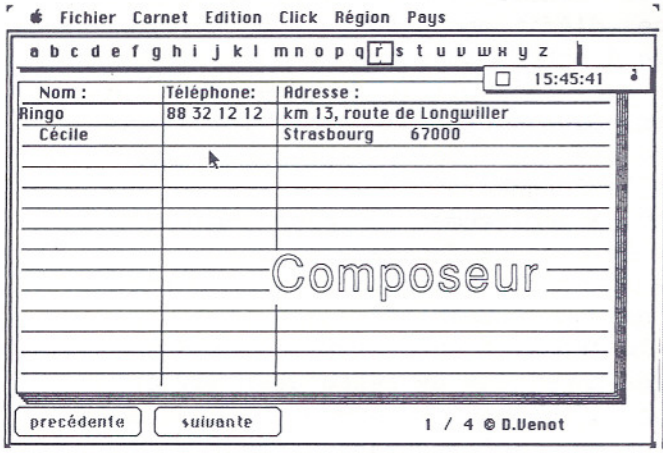
Dans ce qui a précédé, nous avons pris pour exemple le cas d'une application, mais tout ceci est réversible et s'applique aussi aux accessoires de bureau :

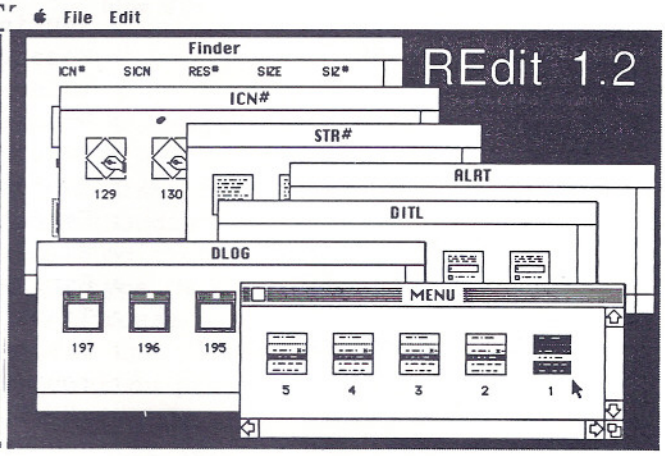
- le paragraphe "démarrage de l'application" correspond à l'ouverture de l'accessoire, provoqué depuis l'application par la routine _OpenDeskAcc (routine Open du driver) ;
- le passage de l'application à l'accessoire correspond ici à un événement *activateEvt* avec le bit *activeFlag* à 1 ;
- le passage de l'accessoire à l'application correspond à un événement *activateEvt* avec le bit *activeFlag* à 0 ;
- la fermeture de l'application correspond à la fermeture de l'accessoire, provoquée depuis l'application par la routine _CloseDeskAcc (routine Close du driver).

# DISQUETTE MACINTOSH 24

Les abonnés aux disquettes "Pom's Macintosh" sont vraiment des veinards : en effet, la disquette 24 comporte un utilitaire indispensable à tous ceux qui programment - ou même utilisent - le Macintosh. Il s'agit de **REdit.** Ce logiciel est un éditeur de ressources évolué qui autorise l'édition directe des icônes, menus, fenêtres, boîtes à dialogue, chaînes de caractères, etc. De plus, c'est aussi un décompilateur capable de faire l'inverse de ce que fait RMaker : vous lui indiquez une ressource, il vous en fait un source au format RMaker. Pour 80,00 F (66,66 F pour les abonnés)... d'autant plus que comme d'habitude, l'acquisition de la disquette d'accompagnement vous évite la saisie des programmes et sources listés dans ce numéro !

ATTENTION : CE LOGICIEL, ENTRE DES MAINS INEXPERTES, PEUT ÊTRE EXTRÊMEMENT "CASSANT" ; IL EST RECOMMANDÉ DE TRAVAILLER UNIQUEMENT SUR DES COPIES.




### *Francophonisme torride ?*

*Font/DA Mover version française a perdu les lettres bjkmoqvwyz ! Les auteurs anglophones avaient en effet trouvé, à titre d'exemple, une phrase courte représentant fort bien l'alphabet : 'The quick brown fox jumps over the lazy dog', phrase malencontreusement traduite par : 'Le renard agile a sauté au-dessus du chien paresseux...', autrement dit il ne s'agit ni d'une traduction littérale, ni d'une représentation complète de l'alphabet... Qui nous donnera un bon équivalent français de cet exemple classique chez les typographes ?*

# LE BUNDLE ET LE FINDER (HISTOIRE TRISTE)

MARIANNE SUTZ

Ce n'est un secret pour personne, le Macintosh, quel que soit le langage de programmation employé, utilise un grand nombre de 'ressources' pour la définition des menus, fenêtres, messages d'erreur, icônes, etc. En fait, presque tout ce qui apparait sur l'écran du Mac est stocké dans des fichiers 'ressource'. Ceci facilite grandement le travail des programmeurs (qui peuvent, par exemple, modifier la taille d'une fenêtre sans toucher au programme lui-même) et des traducteurs (qui utilisent des utilitaires du type "REdit" pour 'localiser' les messages, formats d'affichage des nombres, formats d'affichage des dates et heures...)

Dans les ressources de chaque application, nous trouvons une constante : la partie destinée à informer le "Finder". Nous y trouvons le nom de l'application, le "créateur" et le "type" (nous avons déjà eu l'occasion d'en parler dans de précédents numéros), la signature de l'application, et une ressource de type "Bundle".

Cette dernière, qui contient la liste des identificateurs d'icônes et la liste des références des fichiers, semble justement poser problème à nombre de nos lecteurs. Il est en effet certain que la mise en ordre des types ICN# et FREF n'est pas complètement instinctive. C'est pourquoi nous vous proposons une ressource de base, autour de laquelle vous pourrez construire la ressource de votre application. Que cette dernière soit écrite en assembleur, en Pascal... n'a pas d'importance puisque ces langages sont livrés avec le compilateur de ressource "RMaker", auquel est destiné le fichier listé ci-contre.

## Fichier Démo.R

### Exemple de 'ressource' pour une application

```
Démo                ;;Nom de l'application
APPLPoms            ;;Créateur et type

Type Poms = STR     ;;Signature de l'application
  ,0                ;;Identificateur=0 par convention
* La chaîne suivante contient le nom de l'application,
* le numéro de version, et la date de création.
  Démo - V 2. à la ligne - 22:58:14 - 10.05.86

* Dans un fichier "ressource" destiné à une véritable
* application, nous trouverions ici les ressources de
* type WIND, MENU, DITL, STR, DLOG, etc.

* Ressource de type "Bundle". Elle contient les
* informations pour le Finder.
Type BNDL
   ,128(32)         ;;Identificateur de la ressource
*  (32) est l'attribut de la ressource
*  ce qui correspond à "purgeable"
   Poms 0           ;;Signature et identificateur
*  Identificateurs des icônes représentant l'application
*  et les fichiers qu'elle pourrait créer.
   ICN#
   0 128 1 129 2 130
*  REFérence des Fichiers
   FREF
   0 128 1 129 2 130

Type FREF           ;;Pour l'icône liée à l'application :
   ,128(32)         ;;Type de fichier = APPL
   APPL 0           ;;ID 0 pour la liste des icônes

Type FREF           ;;Pour l'icône du 1er type de fichier :
   ,129(32)         ;;Type de fichier = TEXT
   TEXT 1           ;;ID 1 pour la liste des icônes

Type FREF           ;;Pour l'icône du 2ème type de fichier :
   ,130(32)         ;;Type de fichier = TXT@
   TXT@ 2           ;;ID 2 pour la liste des icônes

* Définition des icônes de l'application et des deux
* types de fichiers générés par cette dernière
TYPE ICN# = GNRL ;;Type liste d'icônes
```

```
* Icône de l'application
  ,128(32)        ;;Identificateur de la liste
2                 ;;Nombre=2 (icône et masque)
00000000 78F3DD86 45099248 78F31044
40030002 41FB7E0C 07FFEF80 0F4BE860
1FFB0490 1DB68498 3FDFF004 3EF8D00C
3ACB0004 7DFD6002 7FDA0042 7FF90006
7BFC000A 7BEC0002 7FEC0026 7FFC4022
7F640004 3FFD101C 3FB38004 3FFE0028
1FB6C028 1FF7D810 0FFCC120 079F98C0
03FB7300 00FF7E00 001FF000 00000000
*
* Masque
FDFFFFCF FDFFFFFF FFFFFFFE FFFFFFFF
FFFFFFFF E7FFFF9F EFFFFFFE 1FFFFFF0
3FFFFFF8 3FFFFFFC 7FFFFFFE 7FFFFFFE
7FFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
FFFFFFFE 7FFFFFFE 7FFFFFFE 7FFFFFFC
3FFFFFFC 3FFFFFF8 1FFFFFF0 0FFFFFE0
07FFFFC0 03FFFF00 00FFFE00 001FF000

* Icône du 1er type de fichier (TEXT)
  ,129(32)        ;;Identificateur de la liste
2                 ;;Nombre=2 (icône et masque)
0FFF0000 08018000 0B014000 0A81E000
0B6EA000 0A6EA000 F8002000 8BF7A000
B8002000 AB7FF800 B8400C00 ABD87FF8
8854400C BBDB580A 8853540F BAC05B75
885ED375 BBC04001 885BDBED B8404001
8FDF5BFD B7C04001 8057DDDD BEC04001
805E5BFD 80404001 FFCFDEFD 00404001
00405EDD 007FC001 00004001 00007FFF
*
* Masque
0FFF0000 0FFF8000 0FFFC000 0FFFE000
0FFFE000 0FFFE000 FFFFE000 FFFFE000
FFFFE000 FFFFF800 FFFFFC00 FFFFFFF8
FFFFFFFC FFFFFFFE FFFFFFFF FFFFFFFF
FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
FFFFFFFF FFFFFFFF FFFFFFFF 007FFFFF
007FFFFF 007FFFFF 00007FFF 00007FFF

* Icône du 2ème type de fichier (TXT@)
  ,130(32)        ;;Identificateur de la liste
2                 ;;Nombre=2 (icône et masque)
0FFF0000 08018000 0BFF4000 0BFFE000
0BFFA000 0BFFA000 F8002000 8BFFA000
BA00A000 BBFFF800 B8400C00 BBDFFFF8
8A5FC00C BBDFDFFA A85FDFFF BBC05FFD
8BDFDFFD BBD04001 ABDFDFFD B8405005
8FDFDFFD BFD04001 BFDFDFFD BFC05005
BFDFDFFD 805FC001 FFDFDFFD 005FDFFD
00405FFD 007FDFFD 00004001 00007FFF
*
* Masque
0FFF0000 0FFF8000 0FFFC000 0E01E000
0E00E000 0FFFE000 FFFFE000 FFFFE000
FFFFE000 EFFFF800 EFFFFC00 FFFFFFF8
FFF07FFC FFF07FFE FFFFF00F FFFFF007
FE7FFFFF FE7FFFFF FFFFFFFF FFFFFFFF
FFFFFFFF FFFFFFFF E07FFFFF E07FFFFF
```

Dans le cas présent, on a supposé que l'application (icône en forme de pomme, type APPL) a la possibilité de générer deux types de fichiers : le type TEXT et le type TXT@ ; vous pouvez bien sûr modifier les types et la forme des icônes, soit en changeant directement les valeurs dans le source, soit, après compilation du source proposé, en éditant les icônes avec un utilitaire comme "REdit".

Si votre application ne génère pas de fichiers, supprimez les types ICN# ID 1 et 2 et les types FREF ID 1 et 2. Si elle ne génère qu'un fichier, supprimez seulement ICN# ID 1 et FREF ID 1.

## Relations entre les types de fichiers et le Finder

Lorsqu'un utilisateur du Macintosh sélectionne un ou plusieurs fichiers et demande 'Ouvrir' ou 'Imprimer' dans le menu 'Fichier' ou, tout simplement, fait un double click sur l'icône d'un fichier, l'application concernée est chargé et doit aller lire, dans une zone que nous allons détailler sans tarder, des informations sur ce qu'a demander l'utilisateur.

La variable globale AppParmHandle ($AEC) contient un Handle (adresse du pointeur) sur une zone de mémoire relogeable contenant les informations recherchées :

```
MOVEA.L   AppParmHandle,A2
MOVEA.L   (A2),A2
TST       (A2)
BNE       Imprimer
SUBQ      #1,2(A2)
BGT       Plusdel
...       ...
...       ...
```

| | |
|---|---|
| message (16 bits) | |
| compteur (16 bits) | |
| n° de réf. du volume (16 bits) | premier fichier |
| type de fichier (32 bits) | |
| n° de version (8 bits) | |
| 8 bits non utilisés | |
| long. du nom de fichier (8 bits) | |
| caractères du nom de fichier | |
| ⋮ | |
| n° de réf. du volume (16 bits) | dernier fichier |
| type de fichier (32 bits) | |
| n° de version (8 bits) | |
| 8 bits non utilisés | |
| long. du nom de fichier (8 bits) | |
| caractères du nom de fichier | |

```
FFFFFFFF FFF07FFF FFF07FFF 007FF007
007FF007 007FFFFF 00007FFF 00007FFF

* Ajout du code objet de l'application
* Le fichier Démo.Code est obtenu avec
* le "Linker" de la manière habituelle :
*
*                [
*
*                Démo
*                /Output Démo.Code
*                /Type 'TEMP'
*
*                $
INCLUDE Démo.Code
```

### Zone 'Message'

- si elle contient 0, l'utilisateur a demandé l'ouverture d'un ou de plusieurs documents ;
- si elle contient 1, l'utilisateur a demandé l'impression d'un ou de plusieurs documents.

### Zone 'Compteur'

Elle contient le nombre de documents à ouvrir ou imprimer. Si cette zone contient 0, cela indique que l'utilisateur a seulement demandé l'ouverture de l'application.

Les zones suivantes apparaissent autant de fois qu'il y a de documents à ouvrir ou à imprimer.

### Zone 'Numéro de référence du volume'

Ce numéro est attribué par le gestionnaire de fichier (routine _MountVol).

### Zone 'Type de fichier'

OSType. Dans le cas d'un programme utilisant les ressources listées ci-contre, il pourrait s'agir du type 'TEXT' ou 'TXT@'

### Zone 'Numéro de version du fichier'

Par convention, toujours 0.

### Zone 'Un octet non utilisé'

### Zone 'Longueur du nom'

Sur un octet.

### Zone 'Nom du fichier'

Chaîne contenant les caractères composant le nom du fichier, plus un caractère nul si le dernier caractère du nom se trouve sur une adresse impaire (dans le cas où le nombre de caractères dans le nom est pair).

## *Compatible incompatible ?*

Voici la dernière liste des problèmes de compatibilité avec le Macintosh Plus répertoriés par Apple et revus par Pom's. Tout d'abord, il faut parler du classement des programmes en cinq catégories.

**Catégorie 0** regroupe les logiciels sans problème, ou possédant une version MacPlus.

**Catégorie 1** logiciels qui nécessitent des System et Finder mis au goût du jour.

**Catégorie 2** logiciels compatibles avec cependant quelques restrictions.

**Catégorie 3** logiciels allergiques à la gestion de fichiers hiérarchique.

**Catégorie 4** les incompatibles.

### Les remèdes...

Pour les catégories 1 et 2 :

• Logiciel non protégé : transférer le système du MacPlus sur une disquette 800Ko, ainsi que tous les fichiers liés au logiciel, et enfin copier à l'aide du F/DA Mover *du MacPlus* les polices et accessoires éventuellement nécessaires.

• Logiciel protégé : démarrer la disquette système du MacPlus et lancer l'application Installer, mettre dans le second lecteur la disquette du logiciel, sélectionner l'option "Macintosh Plus System V1.0" et cliquer dans 'Installer'. S'il n'y a pas assez de place pour effectuer l'adaptation, il vous faut isoler sur une disquette 800Ko le système et le Finder, que vous adaptez comme pour un logiciel non protégé.

Pour la catégorie 3, il convient de travailler sans utiliser de dossiers.

Pour la catégorie 4, procéder comme pour les logiciels protégés (catégories 1 et 2), mais en cliquant dans l'option "Clavier Universel" au lieu de l'option 'Installer' : ainsi, vous adaptez le clavier sans modifier le reste du système.

### Le classement...

**Catégorie 0**

Gestion Clients, Gestion Comptable, Gestion Simil 1.2C, MacListe 2.5, MacTap, Maestria, Medimac, Talkie-Mac version 2.

**Catégorie 1**

4ème dimension version 2.0, ABC Base version 14F, Accessory Pack1, AD MacFichier, Addenda Excel, Airborne, Challenger, Clic Fiche, Click Art, Color Print, Communication Edge, Compta Simil 1.2C, Da Vinci, Esay 3D, Enchanter, Excel, ExperLogo 1.1, FactFinder, Fokker Triplan, Font/DA Mover 3.1, Fontastic, Gestion Devis 2.01, Griffin Terminal, Lode Runner, Logi Font, Mac 2392, MacDraw, Macintosh Pascal 1.0, MacPaint, MacProject, MacSAV, Mac Space, MacSpell+, MacTell II2.1, Mac The Knife (3 disq.), MacWrite, Magic, Management Edge, Megaform 1.0, Microsoft Chart 1.0, Microsoft File, Mind Prober, Modem Universel Apple, Negociation Edge, Paie Mac, PFS (File, Report), Picture Base, Pyramide, Quick Paint, Quick Word, Ready Set Go 2.1, Remember, Sales Edge, Sargon III, SciFont, Silicon Press, SoundCap, StarCross, Stat Works, Suspended, Tech Font, TK! Solver, Transbase, Versaterm, Videoworks, Zork (3 disq.).

**Catégorie 2**

1st Base 2.1, ConcertWare+3.0, CX Base 300, CX Base 500, CX Base M24, DB Master, Fluent Fonts, Jazz, Le Tube ..., Mac Tracks, Mac Vision, Omnis 3, PageMaker 1.1, Switcher 4.4, Telemac Bull, Telemac IBM, ThinkTank, ThunderScan, Word.

**Catégorie 3**

Assembleur 68000, Basic 2.0, Color Chart 2.0, Crunch 1.0, Dollars and Sense, Filevision 1.0, Le Lisp, Lisp Expert Kit, Mac C, MacDent, MacDraft, MacForth, MacFortran, Melusine, Orthogiciel, Overvue 2.0, Slide Show Magician.

**Catégorie 4**

Aztec C 1.06G, ChipWits, ClickOn Worksheet, Compta Mac, MacAdvantage, Mac Agenda, MacCommand, Mac Crypt 1, MacExpert, MacLion, MacManager 1.1, MacPublisher 1.31, MacTerminal 1.0, MacBooster, MegaFiler 1.01, MegaMerge 2.0, Multiplan, MusicWorks, Quartet, Smooth Talker 2.0, Strategic Conquest, Through the Looking Glass, Wizardry et la plupart des jeux non listés dans les catégories précédentes.

Dans le prochain numéro, nous nous attacherons au particularisme des certains programmes, tels que : Basic 2.0, Jazz, MacDraft, MacTerminal 1.0, MacTracks, Microsoft File, Multiplan, Omnis 3,...

# ANALYSE MULTICRITÈRE

*Serge Cattan*

Nous vous proposons un programme d'aide à la décision écrit en Applesoft, et fonctionnant indifféremment sur Apple ][+, //e, //e+ ou //c.

## Théorie

Les méthodes d'analyse multicritère servent à analyser et comparer, en fonction de plusieurs critères, un certain nombre de décisions possibles, que nous appellerons des "projets". Les critères d'évaluation sont souvent conflictuels : ainsi, pour prendre un exemple simple, une entreprise cherche en même temps à maximiser son profit et sa part de marché, ce qui – au moins à court terme – est contradictoire.

Les méthodes d'analyse multicritère utilisent deux types d'information : un jeu de notes (une note pour chaque projet en fonction de chaque critère) et une série de "poids" représentant l'importance relative de chaque critère.

La méthode *Electre II*, sur laquelle repose le programme présenté ici, procède de la façon suivante : tous les projets sont comparés deux à deux, à partir de quoi on obtient deux classements des projets, permettant ainsi une double perspective sur l'ensemble des projets.

### Comparaison de deux projets X et Y

Pour chaque paire de projets X et Y, on calcule :

$P^+(X,Y)$ somme des poids des critères selon lesquels X est meilleur que Y

$P^=(X,Y)$ somme des poids des critères selon lesquels X est aussi bon que Y

$P^-(X,Y)$ somme des poids des critères selon lesquels X est moins bon que Y

X est meilleur que Y pour un critère quand l'écart entre les notes selon ce critère est favorable à X et dépasse le *seuil d'indifférence*. Le seuil d'indifférence d'un critère, notion propre à Electre, est l'écart maximum que l'on peut avoir entre les notes accordées selon ce critère sans que l'on puisse dire que cet égard justifie une préférence en faveur du projet le mieux noté.
On calcule de même un *indice de concordance*.

### Indice de concordance

L'indice de concordance C(X,Y), défini ci-dessous, sera d'autant plus proche de 1 que la préférence vers X par rapport à Y sera plus nette.

$$C(X,Y) = [P^+(X,Y)+P^=(X,Y)] / [P^+(X,Y)+P^=(X,Y)+P^-(X,Y)]$$

On se définit trois seuils de concordance C1, C2 et C3, tels que :

$1 > C1 > C2 > C3 > 0$

les valeurs standards proposées par défaut par le programme Electre II étant 3/4, 2/3 et 3/5.

### Seuils de discordance

On définit de même deux seuils de discordance D1 et D2, tels que :

$0 \leq D1 < D2$

### Surclassement fort

On dira que X surclasse fortement Y si les conditions suivantes sont remplies :

soit
- $P^+(X,Y) \geq P^-(X,Y)$ : le poids des critères selon lesquels X est meilleur que Y dépasse celui des critères où l'inverse est vrai ;
- $C(X,Y) \geq C1$ : il faut atteindre ou dépasser le premier seuil de concordance ;
- pour tout critère où Y est meilleur que X : Note(Y) – Note(X) ≤ D2.

soit
- $P^+(X,Y) \geq P^-(X,Y)$
- $C(X,Y) \geq C2$
- pour tout critère où Y est meilleur que X : Note(Y) – Note(X) ≤ D1.

Il y aura donc surclassement fort dans l'un des deux cas suivants : être plus exigeant sur la concordance, ou plus sévère sur la discordance.

### Surclassement faible

On dira que X surclasse faiblement Y si les conditions suivantes sont remplies :

- $P^+(X,Y) \geq P^-(X,Y)$
- $C(X,Y) \geq C3$
- pour tout critère où Y est meilleur que X : Note(Y) – Note(X) ≤ D2.

En résumé, que le surclassement soit fort ou faible, X surclassera Y si :

- d'une part, le poids cumulé des critères où X est meilleur que Y est suffisamment grand, relativement à celui des critères où l'inverse est vrai ;
- d'autre part, il n'existe pas de critère selon lequel Y est franchement meilleur que X.

### Deux méthodes de classement

Electre II fournit comme résultat deux classements.

Dans le classement direct, les éventualités sont rangées de telle façon qu'une éventualité X précède une éventualité Y si et seulement si X est surclassé par moins d'éventualités que Y.

Dans le classement inverse, les éventualités sont rangées de telle façon qu'une éventualité X précède une éventualité Y si et seulement si X surclasse plus d'éventualités que Y.

Ces deux classements se réfèrent au nombre de surclassements forts. Les ex æquo éventuels sont départagés par le nombre de surclassements faibles.

## Le Programme

Il comprend un menu de cinq rubriques. La première permet de saisir le problème. La seconde permet de retrouver un problème préalablement sauvegardé sur disquette. La troisième autorise la modification d'une donnée. La quatrième donne accès à un mode d'emploi détaillé. Quant à la dernière, elle se charge du retour à l'Applesoft.

## *Programme 'ANALYSE MULTICRITERE'*

*Note : une puce (•) dans une chaîne de caractère symbolise un espace.*

```
100  REM  ************************************
     **********************
110 :
120  REM  CATTAN Serge              ANALYSE MULTI
     CRITERE          09/85
130 :
140  REM ************************************
     **********************
150  ONERR  GOTO 3550
160 T$ = "-------------------------------------
     -----------------------------------"
170 D$ =  CHR$ (4)
180  HOME
190  INVERSE : PRINT  SPC( 40)
200  PRINT "•••••••••••ANALYSE•MULTICRITERE••••
     ••••";
210  PRINT  SPC( 40)
220  NORMAL
230  POKE 34,4: VTAB 8
240  PRINT "ANALYSE•D'UN•PROBLEME••••••••-->•1"
250  PRINT "REPRISE•ANALYSE•EXISTANTE•••-->•2"
260  PRINT "MODIFICATION•DONNEES•••••••••-->•3"
270  PRINT "MODE•D'EMPLOI•••••••••••••••-->•4"
280  PRINT "RETOUR•AU•BASIC••••••••••••••-->•5"
290  POKE 36,11: VTAB 18: INPUT "VOTRE•CHOIX•?•"
     ;R$:R =  VAL (R$)
300  ON R GOTO 350,320,330,4150,340
310  GOTO 290
320  GOSUB 3320: GOTO 680
330  GOSUB 3320: GOSUB 2710: GOTO 680
340  POKE 34,0: END
350  HOME
360  VTAB 8: INPUT "NOM•DE•L'ANALYSE•(?•POUR•LE•
     CATALOGUE)•?";NE$
370  IF NE$ = "?" OR  LEN (NE$) = 0 THEN  GOSUB
     3620: GOTO 350
380  PRINT : INPUT "NOMBRES•D'ACTIONS•••?•";NA$
390 NA =  VAL (NA$): IF NA < 2 THEN  PRINT "•UN•
     CHIFFRE•>•1•S.V.P.": GOTO 380
400  PRINT : INPUT "NOMBRE•DE•CRITERES•?•";NC$:S
     I = 0
410 NC =  VAL (NC$): IF NC < 2 THEN  PRINT "•UN•
     CHIFFRE•>•1•S.V.P.": GOTO 380
420  GOSUB 440: GOTO 490
430  REM =========== DIM TABLEAU ===============
     ====
440  IF ME = 1 THEN  RETURN
450  DIM AC$(NA),CR$(NC),PO(NC)
460  DIM DO(NA,NC),NO(NA,2,2),RA(NA,3),NI(NA),NJ
     (NA),NH(NA)
470  DIM PP(NA,NA),PM(NA,NA),DM(NA,NA),RE(NA,NA,
     2),D1(NC),D2(NC),RS(NA,NA,2)
480 ME = 1: RETURN
490  REM ================================
500  REM  SAISIE =======================
510  HOME
520  FOR I = 1 TO NA
530  PRINT : PRINT "ACTION•N[•";I;"••••••••";: I
     NPUT AC$(I)
540  NEXT I
550  FOR I = 1 TO NC
560  PRINT : PRINT  LEFT$ (T$,40);: PRINT : PRIN
     T "CRITERE•N[•";I;"••••••••";: INPUT CR$(I)
570  INPUT "QUEL•EST•SON•POIDS•?";PO$:PO(I) =  V
     AL (PO$)
580  IF PO(I) = 0 THEN  FLASH : PRINT "UN•CHIFFR
     E•<>•0•S.V.P.": NORMAL : GOTO 570
590 SI = SI + PO(I)
600  NEXT I
610  HOME
620  FOR I = 1 TO NA
630  PRINT : PRINT  LEFT$ (T$,40);: PRINT : INVE
     RSE : PRINT "***•ACTION•";AC$(I);":": NORMA
     L : PRINT
640  FOR J = 1 TO NC
650  PRINT : PRINT "QUEL•EST•SA•NOTE•D'APRES•LE•
     CRITERE•": PRINT CR$(J);"•";: INPUT DO$:DO(
     I,J) =  VAL (DO$)
660  IF DO(I,J) = 0 THEN  FLASH : PRINT "UN•CHIF
     FRE•<>•0•S.V.P.": NORMAL : GOTO 650
670  NEXT J,I
680  REM  CALCUL DES P+ ET DES P- =======
690  REM ================================
700  FOR I = 1 TO NA - 1
710  FOR J = I + 1 TO NA
720 PP(I,J) = 0:PM(I,J) = 0
730  FOR K = 1 TO NC
740 DI = DO(I,K):DJ = DO(J,K)
750  IF DI > DJ THEN PP(I,J) = PP(I,J) + PO(K)
```

```
760  IF DI < DJ THEN PM(I,J) = PM(I,J) + PO(K)
770  NEXT K
780  IF PM(I,J) = 0 THEN PM(I,J) = 0.0001
790  NEXT J,I
800  HOME
810  REM  SAISIE COEFFICIENTS ======
820  REM ==========================
830 BC = 0: IF R = 2 OR R = 3 THEN 850
840  IF BC = 0 THEN 870
850  PRINT : PRINT : INPUT "VOULEZ-VOUS•MODIFIER
     •LES•COEFFICIENTS•••D1•ET•D2•?•";RE$
860  IF  LEFT$ (RE$,1) <  > "O" THEN 950
870  HOME : PRINT "****•SAISIE•DES•SEUILS•DE•DISC
     ORDANCE•****"
880  PRINT  LEFT$ (T$,40);
890  PRINT : PRINT : FOR I = 1 TO NC
900  PRINT "POUR•LE•CRITERE•";CR$(I): PRINT
910  PRINT "D1=•";: INVERSE : PRINT D1(I);"•";:
     NORMAL : INPUT "••";D1$: IF  LEN (D1$) > 0
     THEN D1(I) =  VAL (D1$)
920  PRINT "D2=•";: INVERSE : PRINT D2(I);"•";:
     NORMAL : INPUT "••";D2$: IF  LEN (D2$) > 0
     THEN D2(I) =  VAL (D2$)
930  IF D2(I) =  < D1(I) OR D1(I) =  < 0 THEN  P
     RINT : FLASH : PRINT "IL•FAUT•0•<•D1•<•D2":
      NORMAL : PRINT : GOTO 900
940  NEXT I
950  HOME
960  IF R = 2 OR R = 3 THEN  PRINT : INPUT "VOUL
     EZ-VOUS•MODIFIER•C1,C2,C3•?";RE$: IF  LEFT$
      (RE$,1) <  > "O" THEN 1030
970  HOME : PRINT "****•SAISIE•DES•SEUILS•DE•CONC
     ORDANCE•****"
980  PRINT  LEFT$ (T$,40);
990  PRINT : PRINT "COEFFICIENT•C1•?•";: INVERSE
      : PRINT C1;"•";: NORMAL : INPUT "••";C1$:
     IF  LEN (C1$) > 0 THEN C1 =  VAL (C1$)
1000  PRINT : PRINT "COEFFICIENT•C2•?•";: INVERS
     E : PRINT C2;"•";: NORMAL : INPUT "••";C2$:
      IF  LEN (C2$) > 0 THEN C2 =  VAL (C2$)
1010  PRINT : PRINT "COEFFICIENT•C3•?•";: INVERS
     E : PRIN111818111@~¶111111111111111111181`_
      IF  LEN (C3$) > 0 THEN C3 =  VAL (C3$)
1020  IF C1 =  > 1 OR C2 =  > C1 OR C3 =  > C2 O
     R C3 =  < 0 THEN  PRINT : FLASH : PRINT : P
     RINT "IL•FAUT•0•<•C3•<•C2•<•C1•<•1": NORMAL
      : GOSUB 3120: GOTO 970
1030  REM  RELATIONS DE SURCLASSEMENT ===
1040  REM ==============================
1050  HOME : FLASH : PRINT "••****•LONGUEURS•DES•
     CHEMINS•ISSUS•*****•": NORMAL
1060 K1 = 1:K2 = 2:BC = 1
1070  FOR I = 0 TO NA
1080  FOR J = 0 TO NA
1090  FOR K = K1 TO K2
1100 RE(I,J,K) = 0
1110  NEXT K,J,I
1120  FOR I = K1 TO NA - K1
1130  FOR J = I + K1 TO NA
1140 DM(I,J) = 0:DM(J,I) = 0
1150  FOR K = 1 TO NC
1160 DK = DO(J,K) - DO(I,K)
1170  IF DM(I,J) = K2 THEN 1200
1180  IF DK > D2(K) THEN DM(I,J) = K2: GOTO 1200
1190  IF DK > D1(K) THEN DM(I,J) = K1
1200  IF DM(J,I) = K2 GOTO 1230
1210  IF DK <  - D2(K) THEN DM(J,I) = 2: GOTO 12
     30
1220  IF DK <  - D1(K) THEN DM(J,I) = K1
1230  NEXT K
1240 RP = PP(I,J) / PM(I,J):RM = K1 - PM(I,J) /
     SI: IF RP = 0 AND PM(I,J) = 0.0001 THEN 129
     0
1250  IF RP < K1 GOTO 1410
1260  IF RM < C1 GOTO 1320
1270  IF DM(I,J) = K2 GOTO 1350
1280 RE(I,0,K1) = K1:RE(I,0,K2) = K1:RE(I,J,K1)
     = K1:RE(I,J,K2) = K1:RE(0,J,K1) = K1:RE(0,J
     ,K2) = K1
1290  NEXT J
1300  NEXT I
1310  GOTO 1530
1320  IF RM < C2 GOTO 1370
1330  IF DM(I,J) = 0 GOTO 1280
1340  GOTO 1380
1350  IF RP = K1 GOTO 1410
1360  GOTO 1290
1370  IF RM < C3 GOTO 1350
1380  IF DM(I,J) = K2 GOTO 1350
1390 RE(I,0,K2) = K1:RE(0,J,K2) = K1:RE(I,J,K2)
     = K1
1400  GOTO 1290
1410 RM = K1 - PP(I,J) / SI
1420  IF RM < C1 GOTO 1460
1430  IF DM(J,I) = K2 GOTO 1290
1440 RE(J,0,K1) = K1:RE(J,I,K1) = K1:RE(0,I,K1)
     = K1:RE(J,0,K2) = K1:RE(J,I,K2) = K1:RE(0,I
     ,K2) = K1
1450  GOTO 1290
1460  IF RM < C2 GOTO 1490
1470  IF DM(J,I) = 0 GOTO 1440
1480  GOTO 1290
1490  IF RM < C3 GOTO 1290
1500  IF DM(J,I) = K2 GOTO 1290
1510 RE(J,0,K2) = K1:RE(J,I,K2) = K1:RE(0,I,K2)
     = K1
1520  GOTO 1290
1530  FOR I = 0 TO NA
1540  FOR J = 0 TO NA
1550  FOR K = K1 TO K2
1560 RS(I,J,K) = RE(I,J,K)
1570  NEXT K,J,I
1580 AZ = 1:BZ = 0: GOSUB 2220
1590 NL = 0: HOME
1600  PRINT "LONGUEURS•DES•CHEMINS•ISSUS•:": PRI
     NT : PRINT
1610  FOR I = 1 TO NA
1620 A$ = AC$(NO(I,K1,K1)):GM =  INT (NO(I,K2,K1
     )):NH(I) = NO(I,K1,K1):NI(I) = NO(I,K2,K1)
1630 NL = NL + 1: PRINT : PRINT "POUR•";A$;",•LO
     NGUEURS=•";GM;"•ET•";NO(I,K2,K2):NJ(I) = NO
     (I,K2,K2)
1640  IF NL = 7 AND NA <  > NL THEN NL = 0: GOSU
     B 3120: HOME
1650  NEXT I
1660  GOSUB 3120
1670  HOME : POKE 36,13: FLASH : PRINT•"***•CLAS
     SEMENT•***": NORMAL
1680 RA(K1,K1) = NO(K1,K1,K1)
1690 RA(K1,K2) = K1
```

```
1700  FOR I = K2 TO NA
1710  IF NO(I,K2,K1) < NO(I - K1,K2,K1) GOTO 174
      0
1720 RA(I,K2) = RA(I - K1,K2)
1730  GOTO 1750
1740 RA(I,K2) = I
1750 RA(I,K1) = NO(I,K1,K1)
1760  NEXT I
1770  FOR K = K1 TO K2
1780  FOR I = 0 TO NA
1790  FOR J = 0 TO NA
1800 RE(I,J,K) = RS(J,I,K)
1810  NEXT J,I,K
1820 AZ =  - 1:BZ = NA: GOSUB 2220
1830 NO(0,K2,K1) = 1E30
1840  FOR I = K1 TO NA
1850  IF NO(I,K2,K1) < NO(I - K1,K2,K1) GOTO 187
      0
1860 NO(I,K2,K2) = NO(I - K1,K2,K2): GOTO 1880
1870 NO(I,K2,K2) = I
1880  FOR J = K1 TO NA
1890  IF RA(J,K1) = NO(I,K1,K1) GOTO 1910
1900  NEXT J
1910 RA(J,K2) = RA(J,K2) + NO(I,K2,K2)
1920  NEXT I
1930  FOR I = K1 TO NA - 1
1940  FOR J = I + 1 TO NA
1950  IF RA(I,K2) <  = RA(J,K2) GOTO 1980
1960 Z = RA(I,K1):RA(I,K1) = RA(J,K1):RA(J,K1) =
      Z
1970 Z = RA(I,K2):RA(I,K2) = RA(J,K2):RA(J,K2) =
      Z
1980  NEXT J
1990  NEXT I
2000 CL(K1,K1) = RA(K1,K1)
2010 RA(K1,3) = K1
2020  FOR I = K2 TO NA
2030  IF RA(I,K2) > RA(I - 1,K2) GOTO 2060
2040 RA(I,3) =  -  ABS (RA(I - K1,3))
2050  GOTO 2070
2060 RA(I,3) = I
2070  NEXT I
2080  REM  AFFICHAGE CLASSEMENT =====
2090  REM ==========================
2100  HOME : POKE 36,13: PRINT "***•CLASSEMENT•*
      **": PRINT
2110  PRINT  LEFT$ (T$,40);
2120 NL = 0: FOR I = K1 TO NA
2130 NL = NL + 1: PRINT : PRINT  ABS (RA(I,3));"
      •";AC$(RA(I,K1)):: IF RA(I,3) < 0 THEN  PRI
      NT "•EX•AEQUO";
2140  IF NL > 9 THEN  GOSUB 3120:NL = 0: HOME
2150  NEXT I
2160  VTAB 23: PRINT : INPUT "VOULEZ-VOUS•RECOMM
      ENCER•L'ANALYSE•?";RE$
2170  IF  LEFT$ (RE$,1) = "O" THEN  HOME : GOTO
      840
2180  GOSUB 3140
2190  PRINT : INPUT "VOULEZ-VOUS•IMPRIMER•CES•RE
      SULTATS•?";R$
2200  IF  LEFT$ (R$,1) = "O" THEN  GOSUB 3640
2210  HOME : GOTO 220
2220  REM  CHEMINS LES + LONGS =====
2230  REM ==========================
2240 :
2250  FOR K = K1 TO K2
2260 B = 0
2270  FOR I = K1 TO NA
2280  FOR J = K1 TO NA
2290 RR = RE(I,J,K)
2300  IF RR <  = 0 GOTO 2450
2310 B = K1
2320  IF RE(J,0,K) <  > 0 GOTO 2350
2330 RE(I,J,K) =  - RR
2340  GOTO 2450
2350 I1 = J
2360 B1 = 0
2370  FOR J1 = K1 TO NA
2380 RT = RE(I1,J1,K)
2390  IF RT = 0 GOTO 2430
2400  IF RT > 0 THEN B1 = K1: GOTO 2430
2410 VA = RR - RT
2420  IF VA >  ABS (RE(I,J1,K)) THEN RE(I,J1,K)
      =  - VA
2430  NEXT J1
2440  IF B1 = 0 THEN RE(I,J,K) = 0
2450  NEXT J
2460  NEXT I
2470  IF B = K1 GOTO 2260
2480  NEXT K
2490  FOR K = K1 TO K2
2500  FOR I = K1 TO NA
2510 MM = 0
2520  FOR J = K1 TO NA
2530  IF  ABS (RE(I,J,K)) > MM THEN MM =  ABS (R
      E(I,J,K))
2540  NEXT J
2550 NO(I,K1,K) = I:NO(I,K2,K) = MM
2560  NEXT I
2570  NEXT K
2580  FOR I = K1 TO NA
2590 NO(I,K2,K1) = (NO(I,K2,K1) + NO(I,K2,K2) /
      1000) * AZ + BZ
2600  NEXT I
2610  FOR I = K1 TO NA - K1
2620  FOR J = I + K1 TO NA
2630  IF NO(I,K2,K1) > NO(J,K2,K1) THEN 2680
2640  FOR K = K1 TO K2
2650 Z = NO(I,K1,K):NO(I,K1,K) = NO(J,K1,K):NO(J
      ,K1,K) = Z
2660 Z = NO(I,K2,K):NO(I,K2,K) = NO(J,K2,K):NO(J
      ,K2,K) = Z
2670  NEXT K
2680  NEXT J
2690  NEXT I
2700  RETURN
2710  REM  MODIFICATIONS
2720  REM ==============================
2730  HOME
2740  PRINT : INPUT "VOULEZ-VOUS•MODIFIER•UNE•NO
      TE•?";RE$
2750  IF  LEFT$ (RE$,1) <  > "O" THEN 2900
2760  PRINT "POUR•QUELLE•ACTION•?"
2770  PRINT : INVERSE : FOR I = 1 TO NA: PRINT I
      ;"-";AC$(I): NEXT
2780  NORMAL : PRINT : INPUT A$:A =  VAL (A$)
2790  IF A < I AND A > 0 THEN I = A: GOTO 2810
2800  PRINT : PRINT "ACTION•INEXISTANTE.": GOTO
      2740
2810  PRINT : PRINT "POUR•QUEL•CRITERE•?"
```

```
2820  PRINT : INVERSE : FOR J = 1 TO NC
2830  PRINT J;"-";CR$(J)
2840  NEXT J
2850  NORMAL : PRINT : INPUT C$:C =  VAL (C$)
2860  IF C < J AND C > 0 THEN J = C: GOTO 2880
2870  PRINT : PRINT "CRITERE•INEXISTANT.": GOTO
      2810
2880  PRINT "NOUVELLE•NOTE•?•";: INVERSE : PRINT
      DO(I,J);: NORMAL : PRINT " ";: INPUT DO$:
      IF  LEN (DO$) > 0 THEN DO(I,J) =  VAL (DO$)
2890  REM ----------------------------
2900  PRINT : INPUT "VOULEZ-VOUS•MODIFIER•UNE•AC
      TION•?";RE$
2910  IF  LEFT$ (RE$,1) <  > "O" THEN 3000
2920  PRINT "LISTE•DES•ACTIONS•PRESENTES•:"
2930  PRINT : INVERSE : FOR I = 1 TO NA
2940  PRINT I;"-";AC$(I)
2950  NEXT I
2960  NORMAL : PRINT : INPUT "ACTION•A•MODIFIER•
      ?";A$:A =  VAL (A$)
2970  IF A < I AND A > 0 THEN I = A: GOTO 2990
2980  PRINT : PRINT "ACTION•INEXISTANTE.": GOTO
      2900
2990  NORMAL : PRINT : PRINT : INPUT "NOUVEAU•NO
      M•?";AC$: IF  LEN (AC$) > 0 THEN AC$(I) = A
      C$
3000  PRINT : INPUT "VOULEZ-VOUS•MODIFIER•UN•CRI
      TERE•?";RE$
3010  IF  LEFT$ (RE$,1) <  > "O" THEN 3100
3020  PRINT "LISTE•DES•CRITERES•PRESENTS•:"
3030  PRINT : INVERSE : FOR I = 1 TO NC
3040  PRINT I;"-";CR$(I)
3050  NEXT I
3060  NORMAL : PRINT : INPUT "CRITERE•A•MODIFIER
      •?";A$:A =  VAL (A$)
3070  IF A < I AND A > 0 THEN I = A: GOTO 3090
3080  PRINT : PRINT "CRITERE•INEXISTANT.": GOTO
      3000
3090  NORMAL : PRINT : PRINT : INPUT "NOUVEAU•NO
      M•?";CR$: IF  LEN (CR$) > 0 THEN CR$(I) = C
      R$
3100  RETURN
3110  REM ========================================
      ===
3120  POKE 36,0: VTAB 23: PRINT "APPUYEZ•SUR•UNE
      •TOUCHE•POUR•LA•SUITE.": GET C$
3130  RETURN
3140  REM  SAUVEGARDE SUR DISQUE ===============
      ====================
3150  PRINT
3160  PRINT D$;"OPEN";NE$
3170  PRINT D$;"WRITE";NE$
3180  PRINT NA: PRINT NC
3190  FOR I = 1 TO NA
3200  PRINT AC$(I)
3210  NEXT I
3220  FOR I = 1 TO NC
3230  PRINT CR$(I): PRINT PO(I): PRINT D1(I): PR
      INT D2(I)
3240  NEXT I
3250  PRINT C1: PRINT C2: PRINT C3: PRINT SI
3260  FOR I = 1 TO NA:
3270  FOR J = 1 TO NC
3280  PRINT DO(I,J)
3290  NEXT J,I
3300  PRINT D$;"CLOSE"
3310  RETURN
3320  REM  LECTURE ENREGISTREMENT===============
      ====================
3330  VTAB 23: INPUT "NOM•DE•L'ANALYSE•(?•POUR•L
      E•CATALOGUE)•?";NE$
3340  IF NE$ = "?" OR  LEN (NE$) = 0 THEN  GOSUB
      3620: HOME : GOTO 3330
3350  HOME : INVERSE : POKE 36,8: VTAB 23: PRINT
      "ANALYSE•FICHIER•...";NE$: NORMAL
3360  PRINT
3370  PRINT D$;"OPEN";NE$
3380  PRINT D$;"READ";NE$
3390  INPUT NA: INPUT NC
3400  GOSUB 430
3410  FOR I = 1 TO NA
3420  INPUT AC$(I)
3430  NEXT I
3440  FOR I = 1 TO NC
3450  INPUT CR$(I): INPUT PO(I): INPUT D1(I): IN
      PUT D2(I)
3460  NEXT I
3470  INPUT C1: INPUT C2: INPUT C3: INPUT SI
3480  FOR I = 1 TO NA:
3490  FOR J = 1 TO NC
3500  INPUT DO(I,J)
3510  NEXT J,I
3520  PRINT D$;"CLOSE"
3530  RETURN
3540  REM TRAITEMENT ERREUR====================
      =======
3550  IF  PEEK (222) = 5 THEN  VTAB 10: PRINT "F
      ICHIER•INEXISTANT": PRINT D$;"DELETE";NE$:
      GOSUB 3620: GOTO 3570
3560  PRINT "ERREUR"
3570  FOR IT = 1 TO 1000: NEXT
3580  POKE 34,0: GOTO 180
3590  REM ====================
3600  REM  LECTURE CATALOGUE
3610  REM ===================
3620  PRINT D$;"CATALOG": PRINT : PRINT "APPUYER
      •SUR•UNE•TOUCHE": GET R$: RETURN
3630  REM ============================
3640  REM  LISTING IMPRIMANTE
3650  REM ============================
3660  PRINT D$;"PR#3"
3670 T1$ = "=====================================
      ==========================================
      ==========================================
      ==============="
3680  PRINT : PRINT D$;"PR#1"
3690 N1 = 7:N2 =  INT (NA / N1) + 1: IF NA = N1
      THEN N2 = 1
3700  PRINT  CHR$ (27); CHR$ (64);: PRINT : PRIN
      T  CHR$ (27); CHR$ (14);: REM  GROS CARACTE
      RES
3710  PRINT "ANALYSE•MULTICRITERE•DE•:";NE$
3720  PRINT  CHR$ (15): REM  132 COLONNES
3730 L = 1: FOR K = 1 TO N2
3740  PRINT T1$
3750  PRINT "D1";: POKE 36,7: PRINT "D2";: POKE
      36,12: PRINT "POIDS";: POKE 36,22: PRINT "C
      RITERES";
3760  POKE 36,40: PRINT "ACTIONS•-->";
3770 L1 = L + N1 - 1: IF NA < L1 THEN L1 = NA
```

```
3780 P = 1: FOR J = L TO L1
3790  POKE 36,42 + 12 * P: PRINT  LEFT$ (AC$(J),
     10);".";
3800 P = P + 1
3810  NEXT J
3820  PRINT : PRINT T1$
3830  FOR J = 1 TO NC
3840  PRINT D1(J);: POKE 36,7: PRINT D2(J);: POK
     E 36,13: PRINT PO(J);
3850  POKE 36,19: PRINT  LEFT$ (CR$(J),20);
3860 P = 1: FOR I = L TO L1
3870  POKE 36,45 + 12 * P: PRINT DO(I,J);
3880 P = P + 1
3890  NEXT I
3900  PRINT
3910  NEXT J
3920 L = L + N1: REM  PRINT CHR$(12)
3930  NEXT K
3940  PRINT : PRINT T1$
3950  PRINT "C1 = ";C1;" C2 = ";C2;" C3 = ";C3:
     PRINT
3960  REM ------------------------
3970  PRINT  CHR$ (27); CHR$ (64)
3980  PRINT : PRINT "***•LONGUEUR•DES•CHEMINS•IS
     SUS•***"
3990  PRINT  LEFT$ (T1$,79)
4000  FOR I = 1 TO NA
4010 A$ = AC$(NH(I)):GM =  INT (NI(I))
4020  PRINT "POUR•";A$;: POKE 36,30: PRINT "LONG
     UEURS=•";GM;"•ET•";NJ(I)
4030  NEXT I
4040  REM ------------------------
4050 : PRINT : PRINT : PRINT "***•CLASSEMENT•***
     "
4060  PRINT  LEFT$ (T1$,79)
4070  FOR I = 1 TO NA
4080  PRINT  ABS (RA(I,3));"•:••";AC$(RA(I,K1));
     : IF RA(I,3) < 0 THEN  PRINT "••EX•AEQUO";
4090  PRINT
4100  NEXT I
4110  PRINT  LEFT$ (T1$,79)
4120  PRINT : PRINT D$;"PR#3": PRINT  CHR$ (21):
     PRINT
4130  GOTO 180
4140  RETURN
4150  REM  MODE D'EMPLOI ===========
4160  REM ==========================
4170  PRINT D$;"PR#3": PRINT
4180  INVERSE : PRINT "•••••••••••••••••••ANALYS
     E•MULTICRITERE•••••••••••••••••••••••••••
     •••••••";
4190  NORMAL : PRINT : PRINT : SPEED= 140
4200  PRINT "•••Ce•programme•a•pour•but•de•vous•
     aider•à•faire•le•bon•choix."
4210  PRINT "L'algorithme•développé•ici•est•celu
     i•de•Roy-Bertier•et•Bernard•Besson"
4220  PRINT "appelé•Electre•2•qui•signifie•:•Eli
     mination•et•choix•traduisant•la•réalité."
4230  PRINT : PRINT "Cette•méthode•permet•d'obte
     nir•un•classement•unique•à•partir•de•plusie
     urs"
4240  PRINT "classements•différents•et•souvent•c
     ontradictoires."
4250  PRINT : PRINT "Le•problème•doit•se•présent
     er•sous•forme•d'un•tableau•avec,•en•colonne
     "
4260  PRINT "les•'actions',•et•en•ordonnée•les•c
     ritères.Le•but•sera•de•déterminer•la"
4270  PRINT "meileure•action•compte•tenu•des•cri
     tères."
4280  PRINT : PRINT "A•chaque•critère•sera•affec
     té•2•coefficients•de•divergence•D1•et•D2"
4290  PRINT : PRINT "Afin•de•limiter•les•ex•aequ
     os•vous•choisirez•3•coefficients•de•converg
     ence"
4300  PRINT "baptisés••C1,•C2,•et•C3."
4310  PRINT : PRINT "Ces•coefficients•devront•ré
     pondre•aux•règles•suivantes•:"
4320  PRINT "0<D1<D2••et••0<C3<C2<C1<1•."
4330  PRINT : PRINT "Vous•donnerez•également•un•
     poids•à•chaque•critère.Ce•poids•est•bien•sû
     r"
4340  PRINT "subjectif.Le•mieux•est•de•s'imposer
     •une•échelle•de•valeurs,•de•0•à•20"
4350  PRINT "par•exemple,•les•notes•étant•attrib
     uées•dans•l'ordre•croissant•des"
4360  PRINT "préférences."
4370  PRINT : PRINT "La•tache•est•plus•facile,•l
     orsqu'on•dispose•de•données•chiffrées,•mais
     "
4380  PRINT "étant•donné•que•les•valeurs•les•plu
     s•grandes•sont•attribuées•aux•actions"
4390  PRINT "les•meilleures•il•faudra•dans•le•ca
     s•d'un•cout,•par•exemple,•changer•le"
4400  PRINT "le•signe•afin•d'inverser•l'ordre."
4410  PRINT : PRINT "Prenons•pour•exemple•le•cho
     ix•d'une•voiture:"
4420  PRINT "Les•actions•seraient•les•marques•:G
     OLF•GTI,•RENAULT•5•TURBO•etc."
4430  PRINT "Les•critères•seraient•les•caractéri
     stiques•:VITESSE•MAX,•CONSOMMATION,"
4440  PRINT "PRIX•etc."
4450  PRINT "Les•vitesses•seront•des•chiffres•po
     sitifs,•tandis•que•les•consommations"
4460  PRINT "et•les•prix•seront•des•chiffres•nég
     atifs."
4470  PRINT : PRINT "Si•le•nombre•d'ex•aequo•est
     •trop•important,•vous•pourrez•augmenter•les
     "
4480  PRINT "coefficients•D1,•D2•ou/et•diminuer•
     les•coefficients•C1,•C2,•et•C3"
4490  SPEED= 255
4500  PRINT : PRINT : PRINT "CE•PROGRAMME•A•ETE•
     INSPIRE•PAR•L'ARTICLE•DE•DANIEL•FERRO•PARU•
     DANS•LA"
4510  PRINT "REVUE•SCIENCE•ET•VIE."
4520  PRINT : PRINT "APPUYEZ•SUR•UNE•TOUCHE•POUR
     •RETOUR•AU•MENU.": GET R$
4530  PRINT  CHR$ (21)
4540  GOTO 180
```

*Fichier de démonstration page suivante...*

## Fichier 'VOITURE'

```
7
7
AUDI 100
BMW 520 I
CITROEN CX 25 GTI
FORD SIERRA 2.3
OPEL SENATOR 2L
P. 505 GTI
RENAULT 25 GTX
VITESSE MAXI
4
5
10
1000 M DEPART ARRETE
4
1.5
80 A 120 KM/H EN 5IEME
2
100
200
CONSOMMATION 90 KM/H
2
1
TENUE DE ROUTE
4
1
2
CONFORT
3
1
2
PRIX
5
5000
10000
24
199
-32.3
-505
-6.6
8
6
-132000
190
-31
-585
-7.35
5
9
-117000
192
-31.2
-415
-7.35
7
9
-120000
182
-33
-570
-7.25
6
7
-88000
180
-33
-670
-7.45
5
7
-112000
186
-31.6
-425
-7.05
7
7
-98000
192
-31.5
-380
-6.75
8
8
-105000
5000
```


Maintenant sur //c et //e+

**Basicium...**

**le Basic enrichi**

**Gérard Michel**

Apple ][+, //e, //e+, //c

*Cet utilitaire vous permet de :*

*Gérer des masques 40 ou 80 colonnes. Ces écrans, définis lors de la programmation, s'affichent par la simple commande " ] A".*

*Imprimer l'écran 40 ou 80 colonnes par "]H".*

*Nettoyer mémoire très rapidement par "]F".*

*Gérer des messages, la réponse étant contrôlée.*

*Saisir toutes les variables d'un écran en une seule commande "]I" en contrôlant le type et la longueur.*

*Mais aussi, ne saisir que l'une des variables affectées à un écran, les afficher, les effacer, etc...*

**Tous les fichiers source sont sur la disquette (exemple de démonstration dans le Pom's 13)**

**Disquette et documentation : 150,00 TTC franco (bon de commande page 74)**

**EPE 5.0**

**EPE 5.0, c'est :**

*lister les programmes Basic en avant ET en arrière,*

*modifier, insérer des caractères en plein écran sans relire les lignes,*

*rechercher toute chaîne de caractères,*

**C'est aussi éditer :**

*Sous DOS,*
*Sous ProDOS,*
*En 40 colonnes,*
*En 80 colonnes,*
*Sur Apple //e,*
*Sur Apple //e+,*
*Sur Apple //c*

**EPE 5.0**
**200,00 F TTC, franco**

**Echange EPE ancienne version contre 5.0**
**80,00 F TTC Franco**

# Hard-Copy sur Seïkosha GP500A

Jean-Marc Roulon

Ce programme possédant ses propres compteurs et drapeaux, il peut être associé à n'importe quel autre programme après assemblage à une adresse adéquate. L'appel du programme se fait à ORG + 233 ($E9) et permet une copie "hard" de l'une des 2 pages graphiques après initialisation des adresses suivantes :

- **ORG + 2 :**
  mode NORMAL($00) ou INVERSE($01)
- **ORG + 3 :**
  impression SIMPLE($00) ou DOUBLE($01)
- **ORG + 4 :**
  SANS CADRE($00) ou AVEC CADRE($01)
- **ORG + 5 :**
  POSITION du dessin par rapport au bord gauche du papier. De 0 à 255 ($00 à $FF) en impression simple et de 0 à 50 ($00 à $32) en impression double.
- **ORG + 6 :**
  PAGE HGR1($20) ou PAGE HGR2($40)

IMPORTANT : il ne faut pas configurer les ports 'série' par l'utilitaire système, le programme initialisant lui-même la sortie imprimante (slot 1).

## Source 'T.DHARD'

### Assembleur BIG MAC

```
 1 **************************
 2 *                        *
 3 * DOUBLE.HARD.COPIE.HGR  *
 4 * POUR SEIKOSHA GP-500AS *
 5 *     SUR APPLE//C       *
 6 *                        *
 7 *                        *
 8 *     PAR J-M ROULON     *
 9 *                        *
10 *  LE 23/JANVIER/1986    *
11 *                        *
12 **************************
13
14
15 *******************
16 *   LES ADRESSES  *
17 *******************
18 OUTPORT  EQU  $FE95
19 COUT     EQU  $FDED
20 NBCAR    EQU  $07F9
21
22          ORG  $6000
23
24 HGRTAMP  DS   $01
25 TAMP     DS   $01
26 INVERSE  DS   $01
27 DOUBLE   DS   $01
28 CADRE    DS   $01
29 POSITION DS   $01
30 PAGE     DS   $01
31 COMPT1   DS   $01
32 COMPT2   DS   $01
33 COMPT3   DS   $01
34 COMPT4   DS   $01
35 COMPT5   DS   $01
36 ADR1     DS   $01
37 ADR2     DS   $D1
38 INITCAR  HEX  005032090844
                 3509503709
39 ************************
40 *    INITIALISATION    *
41 *     DES COMPTEURS    *
42 ************************
43          JSR  DLIGNE
44          JSR  SUITE5-5
45          JSR  DCADRE
46          LDA  #$D0
47          STA  COMPT5
48 DEBUT    STA  HGRL
49          JSR  DLIGNE
50          JSR  DPOS
51          LDA  #$00
52          JSR  COUT
53          LDY  #$00
54          LDX  #$00
55          LDA  #$03
56          STA  COMPT1
57 B1       LDA  PAGE
58          STA  HGRH
59          STA  HGRTAMP
60          LDA  #$08
61          STA  COMPT2
62 B2       LDA  #$08
63          STA  COMPT3
64 **********************
65 *  PROGRAMME-MAITRE  *
66 **********************
67 PGRM     HEX  AD      "LDA"
68 HGRL     HEX  00
69 HGRH     HEX  00
70          CPX  INVERSE
71          BEQ  SUITE
72          EOR  #$FF
73 SUITE    AND  #$7F
74          CPX  DOUBLE
75          BEQ  AA
76          JSR  DHARD
77 AA       JSR  COMPAR
78          JSR  COUT
79          STX  NBCAR
80          LDA  HGRH
81          SEC
82          SBC  #$04
83          STA  HGRH
84          DEC  COMPT3
85          BNE  PGRM
86          LDA  HGRTAMP
87          STA  HGRH
88          LDA  HGRL
89          CLC
90          ADC  #$80
91          STA  HGRL
92          BCS  BB
93          DEC  HGRH
94 BB       LDA  HGRH
95          STA  HGRTAMP
96          DEC  COMPT2
97          BNE  B2
98          LDA  HGRL
99          SEC
100         SBC  #$28
101         STA  HGRL
102         DEC  COMPT1
103         BNE  B1
104         JSR  FLIGNE
105         LDX  #$00
106         CPX  DOUBLE
107         BEQ  SUITE3
108         JSR  DLIGNE
109         JSR  DPOS
110         LDA  #$00
111         JSR  COUT
112
113 LIGNE2  LDA  ADR2,Y
114         AND  #$7F
115         JSR  COMPAR
116         JSR  COUT
117         JSR  COUT
118         STX  NBCAR
119         INY
120         CPY  #$C0
121         BNE  LIGNE2
122         JSR  FLIGNE
```

*Ce source, sur la disquette d'accompagnement, est sauvegardé en format TEXT pour permettre l'édition avec un autre assembleur.*

```
123 SUITE3   INC  COMPT5
124          LDA  COMPT5
125          CMP  #$F8
126          BEQ  FIN
127          JMP  DEBUT
128 FIN      JSR  DCADRE
129          RTS
130 ************************
131 *     SOUS-PROGRAMME   *
132 * DOUBLE HARD.COPY.HGR *
133 ************************
134 DHARD    STA  TAMP
135          LDA  #$00
136          STA  ADR1
137          CLC
138          LDX  #$04
139          STX  COMPT4
140 DEB1     DEC  COMPT4
141          BEQ  SUITE1
142          ROR  TAMP
143          BCC  C0
144          ROR  ADR1
145          SEC
146          ROR  ADR1
147          JMP  DEB1
148 C0       ROR  ADR1
149          CLC
150          ROR  ADR1
151          JMP  DEB1
152 SUITE1   ROR  TAMP
153          BCC  SC0
154          ROR  ADR1
155          SEC
156          ROR
157          JMP  DEB2
158 SC0      ROR  ADR1
159          CLC
160          ROR
161 DEB2     STX  COMPT4
162 DEB3     DEC  COMPT4
163          BEQ  RETURN
164          ROR  TAMP
165          BCC  C20
166          ROR
167          SEC
168          ROR
169          JMP  DEB3
170 C20      ROR
171          CLC
172          ROR
173          JMP  DEB3
174 RETURN   CLC
175          ROR
176          CLC
177          ROR  ADR1
178          STA  ADR2,Y
179          INY
180          LDX  #$00
181          LDA  ADR1
182          AND  #$7F
183          JSR  COMPAR
184          JSR  COUT
185          RTS
186 *************************
187 *      COMPARAISON      *
188 * AVEC CHR$(9)&CHR$(13) *
189 *************************
190 COMPAR   CMP  #$09
191          BEQ  CC
192          CMP  #$0D
193          BNE  DD
194 CC       ADC  #$01
195 DD       RTS
196 ************************
197 *    INITIALISATION    *
198 *   DE L'IMPRIMANTE    *
199 ************************
200 DLIGNE   LDA  #$01
201          JSR  OUTPORT
202          LDX  #$0A
203 BLIGNE   LDA  INITCAR,X
204          JSR  COUT
205          DEX
206          BNE  BLIGNE
207          RTS
208 ********************
209 *     POSITION     *
210 *        ET        *
211 *      CADRE       *
212 ********************
213 DPOS     LDX  POSITION
214 BPOS     BEQ  SUITE4
215          LDA  #$00
216          JSR  COUT
217          STA  NBCAR
218          DEX
219          JMP  BPOS
220 SUITE4   CPX  CADRE
221          BEQ  RETOUR
222          LDX  #$07
223          JSR  BCADRE
224 RETOUR   RTS
225 *****************
226 *  FIN DE LIGNE  *
227 *****************
228 FLIGNE   LDA  #$00
229          JSR  COUT
230          CPX  CADRE
231          BEQ  SUITE5
232          LDX  #$07
233          JSR  BCADRE
234          LDA  #$00
235          JSR  COUT
236 SUITE5   LDA  #$09
237          JSR  COUT
238          LDA  #$37
239          JSR  COUT
240          LDA  #$50
241          JSR  COUT
242          LDA  #$0A
243          JSR  COUT
244          LDA  #$0D
245          JSR  COUT
246          LDA  #$00
247          JSR  OUTPORT
248          RTS
249 *****************
250 *   LE CADRE    *
251 *****************
252 DCADRE   LDX  #$00
253          CPX  CADRE
254          BEQ  RETURN2
255          JSR  DLIGNE
256          JSR  DPOS
257          LDX  #194
258          JSR  BCADRE
259          CPX  DOUBLE
260          BEQ  FLIGNE+5
261          LDX  #192
262          JSR  BCADRE
263          JMP  FLIGNE+5
264 RETURN2  RTS
265 BCADRE   LDA  #$7F
266          JSR  COUT
267          LDY  #$00
268          STY  NBCAR
269          DEX
270          BNE  BCADRE
271          RTS
272          END
```

## Programme 'COPIEHGR'

*Les • symbolisent des espaces.*

```
5  HOME :T$(1) = "OPTIONS D'IMPRESSION":T$
   (2) = "--------------------":A$(1) = "I
   MPRIMER UNE PAGE GRAPHIQUE":A$(2) = "CH
   ARGER UNE PAGE GRAPHIQUE":A$(3) = "VOIR
    UNE PAGE GRAPHIQUE":A$(4) = "LARGEUR D
   'IMPRESSION :"
10 A$(5) = "NUMERO DU LECTEUR••••:":A$(6)
   = "MODE D'IMPRESSION••••:":A$(7) = "NUM
   ERO DE PAGE•••••••:":A$(8) = "AVEC CADR
   E•••••••••••:":A$(9) = "POSITION   (0/•
   ••••••:":OP$(6) = "125": PRINT  CHR$ (4
   );"BLOAD DHARD"
20  DATA  8,9,2,4,6,11,13,15,17,19,21
30  FOR I = 1 TO 2: READ V: HTAB 12: VTAB
   V: PRINT T$(I);: NEXT I: FOR I = 1 TO 9
   : READ V: HTAB 3: VTAB V: INVERSE : PRI
   NT I;: NORMAL : PRINT " - ";A$(I): NEXT
    I: RESTORE
40  GOSUB 80: GOSUB 400: GOSUB 500: GOSUB
   600: GOSUB 700: GOSUB 800: GOSUB 910: G
   OSUB 70
50 X =  PEEK ( - 16384) - 176: IF X < 1 OR
    X > 9 THEN 50
60  POKE  - 16368,0: ON X GOSUB 100,200,30
```

```
0,400,500,600,700,800,900: GOSUB 70: GO
TO 50
70 V = 11: FOR I = 1 TO 6: HTAB 30: VTAB V
: PRINT OP$(I);"••":V = V + 2: NEXT I:
HTAB 21: VTAB 21: PRINT LP$;: RETURN
80  FLASH : HTAB 5: VTAB 24: PRINT "APPUYE
R SUR LE NUMERO DESIRE";: NORMAL : RETU
RN
100  CALL 24809: RETURN
200  POKE 34,23: HOME : INPUT "•••NOM ?:";
N$: PRINT  CHR$ (4);"BLOAD ";N$;",D";
VAL (OP$(2));",A$";AD: POKE 34,0: GOSU
B 80: GOSUB 300: RETURN
300  POKE  - 16304,0: POKE  - 16297,0: POK
E  - 16302,0: POKE  - 16349,0: POKE  -
16349 +  ASC (OP$(4)),0: GET R$: TEXT
: RETURN
400  IF IM = 1 THEN OP$(1) = "DOUBLE": POK
E 24579,1:IM = 0:LP$ = "80)•": RETURN
410 OP$(1) = "SIMPLE": POKE 24579,0:IM = 1
:LP$ = "255)": RETURN
500  IF L = 1 THEN OP$(2) = "2":L = 0: RET
URN
510 OP$(2) = "1":L = 1: RETURN
600  IF M = 1 THEN OP$(3) = "INVERSE": POK
E 24578,1:M = 0: RETURN
610 OP$(3) = "NORMAL ": POKE 24578,0:M = 1
: RETURN
700  IF P = 1 THEN OP$(4) = "2": POKE 2458
2,95:P = 0:AD = 4000: RETURN
710  POKE 24582,63:P = 1:OP$(4) = "1":AD =
2000: RETURN
800  IF C = 1 THEN OP$(5) = "OUI": POKE 24
580,1:C = 0: RETURN
810 OP$(5) = "NON": POKE 24580,0:C = 1: RE
TURN
900  POKE 34,23: HOME : INPUT "•••POSITION
?:";OP$(6): POKE 34,0: GOSUB 80
910  POKE 24581, VAL (OP$(6)): RETURN
```

## Récapitulation 'DHARD'

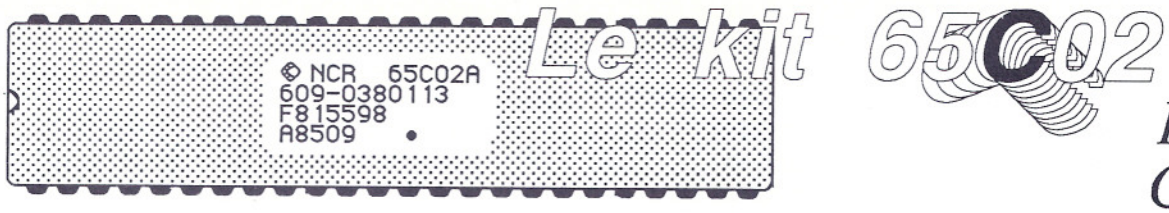
*Après avoir saisi ce code sous moniteur, vous le sauvegarderez par BSAVE DHGR,A$6000,L$2C0*

```
6000- FF FF 00 00 FF FF 00 00
6008- FF FF 00 00 FF FF 00 00
6010- FF FF 00 00 FF FF 00 00
6018- FF FF 00 00 FF FF 00 00
6020- FF FF 00 00 FF FF 00 00
6028- FF FF 00 00 FF FF 00 00
6030- FF FF 00 00 FF FF 00 00
6038- FF FF 00 00 FF FF 00 00
6040- FF FF 00 00 FF FF 00 00
6048- FF FF 00 00 FF FF 00 00
6050- FF FF 00 00 FF FF 00 00
6058- FF FF 00 00 FF FF 00 00
6060- FF FF 00 00 FF FF 00 00
6068- FF FF 00 00 FF FF 00 00
6070- FF FF 00 00 FF FF 00 00
6078- FF FF 00 00 FF FF 00 00
6080- FF FF 00 00 FF FF 00 00
6088- FF FF 00 00 FF FF 00 00
6090- FF FF 00 00 FF FF 00 00
6098- FF FF 00 00 FF FF 00 00
60A0- FF FF 00 00 FF FF 00 00
60A8- FF FF 00 00 FF FF 00 00
60B0- FF FF 00 00 FF FF 00 00
60B8- FF FF 00 00 FF FF 00 00
60C0- FF FF 00 00 FF FF 00 00
60C8- FF FF 00 00 FF FF 00 00
60D0- FF FF 00 00 FF FF 00 00
60D8- FF FF 00 00 FF FF 00 50
60E0- 32 09 08 44 35 09 50 37
60E8- 09 20 32 62 20 6E 62 20
60F0- 92 62 A9 D0 8D 0B 60 8D
60F8- 22 61 20 32 62 20 43 62
6100- A9 00 20 ED FD A0 00 A2
6108- 00 A9 03 8D 07 60 AD 06
6110- 60 8D 23 61 8D 00 60 A9
6118- 08 8D 08 60 A9 08 8D 09
6120- 60 AD 00 00 EC 02 60 F0
6128- 02 49 FF 29 7F EC 03 60
6130- F0 03 20 B8 61 20 27 62
6138- 20 ED FD 8E F9 07 AD 23
6140- 61 38 E9 04 8D 23 61 CE
6148- 09 60 D0 D5 AD 00 60 8D
6150- 23 61 AD 22 61 18 69 80
6158- 8D 22 61 B0 03 CE 23 61
6160- AD 23 61 8D 00 60 CE 08
6168- 60 D0 B1 AD 22 61 38 E9
6170- 28 8D 22 61 CE 07 60 D0
6178- 95 20 5F 62 A2 00 EC 03
6180- 60 F0 24 20 32 62 20 43
6188- 62 A9 00 20 ED FD B9 0D
6190- 60 29 7F 20 27 62 20 ED
6198- FD 20 ED FD 8E F9 07 C8
61A0- C0 C0 D0 EA 20 5F 62 EE
61A8- 0B 60 AD 0B 60 C9 F8 F0
61B0- 03 4C F7 60 20 92 62 60
61B8- 8D 01 60 A9 00 8D 0C 60
61C0- 18 A2 04 8E 0A 60 CE 0A
61C8- 60 F0 19 6E 01 60 90 0A
61D0- 6E 0C 60 38 6E 0C 60 4C
61D8- C6 61 6E 0C 60 18 6E 0C
61E0- 60 4C C6 61 6E 01 60 90
61E8- 08 6E 0C 60 38 6A 4C F6
61F0- 61 6E 0C 60 18 6A 8E 0A
61F8- 60 CE 0A 60 F0 11 6E 01
6200- 60 90 06 6A 38 6A 4C F9
6208- 61 6A 18 6A 4C F9 61 18
6210- 6A 18 6E 0C 60 99 0D 60
6218- C8 A2 00 AD 0C 60 29 7F
6220- 20 27 62 20 ED FD 60 C9
6228- 09 F0 04 C9 0D D0 02 69
6230- 01 60 A9 01 20 95 FE A2
6238- 0A BD DE 60 20 ED FD CA
6240- D0 F7 60 AE 05 60 F0 0C
6248- A9 00 20 ED FD 8D F9 07
6250- CA 4C 46 62 EC 04 60 F0
6258- 05 A2 07 20 B2 62 60 A9
6260- 00 20 ED FD EC 04 60 F0
6268- 0A A2 07 20 B2 62 A9 00
6270- 20 ED FD A9 09 20 ED FD
6278- A9 37 20 ED FD A9 50 20
6280- ED FD A9 0A 20 ED FD A9
6288- 0D 20 ED FD A9 00 20 95
6290- FE 60 A2 00 EC 04 60 F0
6298- 18 20 32 62 20 43 62 A2
62A0- C2 20 B2 62 EC 03 60 F0
62A8- BB A2 C0 20 B2 62 4C 64
62B0- 62 60 A9 7F 20 ED FD A0
62B8- 00 8C F9 07 CA D0 F3 60
```

**Comment faire ?**

*On ne peut 'BRUNer' ce programme. Dans la pratique, pour imprimer la page graphique 1 en normal, double, sans cadre, à gauche, vous ferez :*

```
ORG=24576  (=$6000)
BLOAD  DHGR,A24576
POKE  ORG+2,0
POKE  ORG+3,1
POKE  ORG+4,0
POKE  ORG+5,0
POKE  ORG+6,32
CALL  ORG+233
```




*Pascal Cantot*

Les possesseurs d'un Apple //e ancien modèle (avec microprocesseur 6502) qui souhaitent le moderniser et le rendre compatible avec le //c peuvent acheter un "kit de mise à niveau 65C02". Il leur en coutera 1000,00 F environ, mais après, ô! merveille...

Épistole //c et compagnie vous montrent plein de petites icônes en mode texte, et, pour les assemblistes invétérés, ProCODE vous permet d'employer 27 nouvelles instructions, et 2 nouveaux modes d'adressage en langage machine, renouvelant ainsi le plaisir de la programmation en assembleur.

Tout serait pour le mieux dans le meilleur des mondes possibles si toutefois la 'documentation' fournie avec le kit 65C02 dépassait le stade de l'installation de ce dernier... Le but de cet article est de faire le tour des très intéressantes possibilités de ces nouvelles ROMs. Notez que cela concerne aussi les possesseurs de //c... Pour la liste des différences entre les ROMs Basic et F8, vous pouvez vous reporter à l'article d'Yvan Koenig publié dans le numéro 21 de Pom's.

Commençons par le microprocesseur. Celui-ci est maintenant un CMOS 6502, ou 65C02 (Métal / Oxyde / Silicium, alimentation Complémentaire), technologie permettant une faible consommation (4 mA).

## Nouvelles instructions

- **BRA** (BRanch Always) : branchement inconditionnel. Prend 2 cycles (3 si changement de page).
- **DEA** (DEcrement Accumulator) : retranche 1 au contenu de l'accumulateur. Agit sur N et Z. Durée : 2 cycles.
- **INA** (INcrement Accumulator) : ajoute 1 au contenu de l'accumulateur. Agit sur N et Z. 2 cycles.
- **PHX** (PusH X on stack) : met le registre X sur la pile. N'affecte pas le registre d'état. Prend 3 cycles.
- **PHY** (PusH Y on stack) : met le registre Y sur la pile. 3 cycles.
- **PLX** (PulL X from stack) : met l'octet situé au sommet de la pile dans X et décrémente le pointeur de pile S. Prend 4 cycles, et agit sur N et Z.
- **PLY** (PulL Y from stack) : met l'octet situé au sommet de la pile dans Y et décrémente S. 4 cycles. Agit sur N et Z.
- **STZ** (STore Zero) : met à 00 l'octet situé à l'adresse spécifiée. N'affecte pas le registre d'état. "STZ abs" prend 4 cycles, "STZ zpg" 3 cycles, "STZ zpg,X" 4 cycles, "STZ abs,X" 5 cycles.
- **TRB** (Test and Reset memory Bits with accumulator, ouf !) : les bits 6 et 7 de la mémoire sont copiés dans V et N (comme pour BIT), puis les bits à 1 dans l'accumulateur mettent à zéro les bits de même rang dans la mémoire (M<-A^M). Par exemple, si A=%10010111 et ADR contient %10101010, TRB ADR donne N=1, V=0, ADR contient %00101000. TRB agit sur N,V,Z. "TRB abs" prend 6 cycles et "TRB zpg" 5 cycles.
- **TSB** (Test and Set memory Bits with accumulator) : les bits 6 et 7 de la mémoire sont copiés dans V et N, puis les bits à 1 de l'accumulateur mettent à 1 les bits correspondants de la mémoire (M<-AvM). N,V,Z sont modifiés. "TSB abs" prend 6 cycles et "TSB zpg" 5 cycles.

## Nouveaux modes d'adressage

### Indirect page zéro

Exemple : LDA ($06) charge l'accumulateur avec le contenu de l'adresse qui se trouve au double-octet (ordre L,H) spécifié. C'est équivalent à "LDY #$00 puis LDA (zpg),Y", mais nettement plus pratique ! Avec ce mode d'adressage, CMP prend 2 cycles, et ADC, AND, EOR, LDA, ORA, SBC, STA, 5 cycles. Par exemple, s'il y a $01 en $B8 et $08 en $B9, "LDA ($B8)" exécutera "LDA $0801".

**Les 27 nouveaux codes**

```
04:TSB zpg
1a:INA
3A:DEA
62:ADC (zpg)
7C:JMP abs  (ind,X)
9C:STZ abs
DA:PHX
0C:TSB abs
1C:TRB abs
3C:TRB abs,X
64:STZ zpg
80:BRA
9E:STZ abs,X
F2:SBC (zpg)
12:ORA (zpg)
32:AND (zpg)
42:EOR (zpg)
74:STZ zpg,X
89:BIT imm
B2:LDA (zpg)
FA:PLX
14:TRB zpg
34:BIT zpg,X
5A:PHY
7A:PLY
92:STA (zpg)
D2:CMP (zpg)
```

### Indirect indexé absolu

Ne s'applique qu'à l'instruction JMP abs(ind,X). X est ajouté à 'ind', et on effectue un JMP à l'adresse contenue dans le mot de 16 bits pointé par le résultat. Ce mode d'adressage est très pratique pour gérer des tables d'adresses de sous-programmes, par exemple dans les amper-interpréteurs... Cette instruction JMP prend 6 cycles.

Notons qu'à la différence du 6502, dans le cas d'un JMP indirect du type "JMP ($xxFF)", l'adresse de la page augmente d'un et cela ajoute un cycle à la durée de l'instruction 65C02 (ce n'était pas le cas avec le 6502).

## Autres différences

Voilà pour les nouvelles instructions. Mais ce n'est pas tout... En effet, la gestion des interruptions a quelque peu changée : avec le 6502, après un BRK, le vecteur d'interruption est chargé dans le compteur ordinal et le vecteur BRK est ignoré. Avec le 65C02, le BRK est exécuté, puis l'interruption est exécutée.

Enfin, une dernière différence pouvant poser quelques problèmes : les codes inutilisés. En effet, il y a 256 codes d'instructions machine possibles, mais seulement 178 sont utilisés. Avec le 6502, les codes inconnus étaient interprétés de façon hasardeuses. Certains donnaient de nouvelles instructions (Ce bug a d'ailleurs été exploité par Apple à l'adresse $FAF4 de la ROM de l'ancien //e), d'autres plantaient carrément l'Apple. Avec le 65C02, toutes les instructions inconnues sont interprétées comme des NOP (pas d'opération). Mais, selon le code, l'instruction aura 1, 2 ou 3 octets et durera de 1 à 8 cycles :

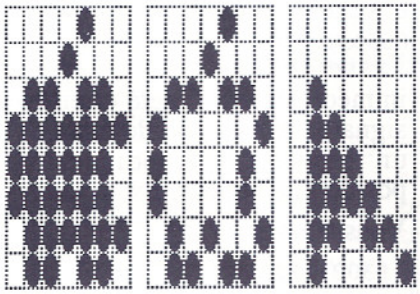
| CODE | OCTETS | CYCLES |
|---|---|---|
| X2 | 2 | 2 |
| X3,X7,XB,XF | 1 | 1 |
| 44 | 2 | 3 |
| 54,D4,F4 | 2 | 4 |
| 5C | 3 | 8 |
| DC,FC | 3 | 4 |

*X = chiffre hexadécimal quelconque*

Si vous ne recherchez pas la compatibilité totale avec les versions passées et à venir du 6502, vous pouvez utiliser ces pseudo-NOPs comme temporisateurs (par exemple dans une routine de 'timing' pour l'écriture d'octets sur une disquette).

Et voilà ! Nous avons fait le tour des principales différences du 65C02 par rapport au 6502. Notons qu'il y en a d'autres de moindre importance, comme la gestion du mode décimal et certains protocoles 'hardware'...

## La ROM vidéo



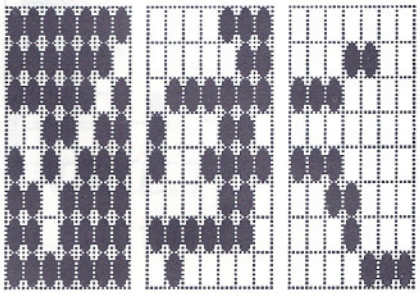
Plusieurs modifications ont été apportées. Tout d'abord, vous avez maintenant à votre disposition 32 caractères graphiques, qui sont décrits à la page 4 du manuel du kit. L'envoi d'un

```
CHR$(27)
```

commute (à condition évidemment que la carte 80 colonnes soit active, ou encore que le jeu de caractères alternés soit commuté par une écriture à l'adresse $C00F) ces caractères à la place des majuscules inversées (codes $40 à $5F), d'où les signes étranges qui parasitent la ligne d'état de votre traitement de texte favori, ou le curseur du tableur si indispensable à votre travail...
Pour obtenir ces caractères, essayez le petit programme suivant :

```
10 PRINT CHR$(4)"PR#3"
20 PRINT CHR$(27): INVERSE:
   REM Commute l'affichage
   des icones
30 FOR I = 64 TO 95: PRINT
   I;" - ";CHR$(I),
40 NEXT: PRINT: NORMAL
50 PRINT CHR$(24): REM
   Revient au mode
   d'affichage normal.
```

Joli, n'est-ce pas ? Mais il y a deux problèmes qui se posent : impérativement, dans un programme machine ayant ses propres routines d'affichage, les majuscules inverses doivent utiliser les codes écran $00 à $1F... De plus, la totalité des programmes de hardcopy d'écran texte risquent de donner des résultats bizarres avec un écran plein d'icônes...



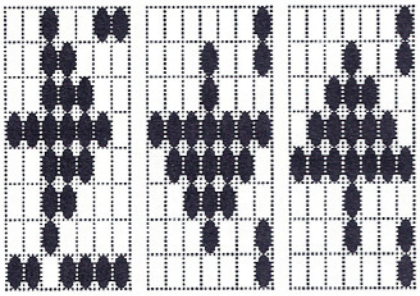
Les modifications ne s'arrêtent pas là. La commande ESC R a été supprimée (vous en comprendrez la raison plus loin), et on a implanté une nouvelle commande : ESC D (tapée AU CLAVIER) permet de filtrer les caractères de contrôle envoyés à l'écran, exceptés CTRL-G (bell), CTRL-H (backspace), CTRL-J (line feed), CTRL-M (carriage return, si, si !). C'est très utile par exemple lorsque l'on reçoit des codes hétéroclites via l'interface série, et que l'on veut éviter qu'il y ait le fouillis sur l'écran (passage de 80 en 40 colonnes, passage en inverse, etc...). On revient au mode normal par ESC E.

Également, pour ceux qui n'auraient pas remarqué, l'affichage en 80 colonnes a été notablement accéléré (30%), et le vilain 'scrolling' qui autrefois rendait une ligne de programme illisible du fait de son défilement en zig-zag est maintenant impeccable.

De plus, sous Pascal UCSD, en envoyant un CTRL-F à l'écran,

on supprime l'affichage du curseur, ce qui permet d'accélérer l'affichage sous Pascal de 30 autres % (il en a bien besoin !). CTRL-E fait revenir le sympathique rectangle sur votre moniteur...

COUT1 ($FDF0) et KEYIN ($FD1B), qui n'étaient pas compatibles avec le mode 80 colonnes, le sont désormais. La routine SETVID ($FE93) déconnecte maintenant l'affichage 80 colonnes.



## Le moniteur

Il accepte désormais TOUTES les commandes en minuscules comme en majuscules. De nouvelles fonctions sont apparues :

### Entrée de codes ASCII

Il est possible d'entrer des codes ASCII à la place d'un ou plusieurs octets en écrivant ('*) (où * est le caractère) à la place des deux chiffres hexadécimaux. Par exemple "400: 'B 'A C1 C2" fera apparaître "BABA" en haut de l'écran (s'il n'y a pas de 'scrolling' bien sûr).

### Recherche d'un ou de deux octets

Pour rechercher le nombre de 16 bits LLHH entre les adresses "début" et "fin", on écrit "HHLL<début.fin" (notez l'inversion de HH et de LL). Par exemple, "'K'O<801.4000S" recherchera "OK". Si on ne spécifie qu'un seul octet, ce sera cet octet et seulement lui qui sera recherché. Ainsi, "60<F800.FFFF" affichera l'adresse de tous les RTS de la ROM F8.

### Mini-assembleur

Mais oui, vous ne rêvez pas ! Il est revenu... et c'est toujours le même depuis 10 ans. Autrement dit : il n'assemble pas, hélas, les codes mnémoniques 65C02. Tout comme la fonction 'List' du nouveau moniteur du //e ne désassemble pas les codes 65C02... Apple a apporté de très intéressantes améliorations, mais aurait pu faire un effort de ce côté !

Bon, c'est tout de même mieux que rien. Les nouvelles instructions du 65C02 ne sont pas irremplaçables... Pour entrer dans le mini-assembleur, vous tapez "!". Le prompt du moniteur (*) se change en (!). La syntaxe est la suivante: "HHLL:instr". Si, à la place de l'adresse d'assemblage HHLL, vous tapez un espace, le code sera implanté à la suite de la dernière instruction assemblée. Le résultat de l'assemblage est affiché immédiatement. Toute erreur est signalée par un son de cloche et un pointeur (^) indiquera la position de l'erreur. Exemple :

Vous tapez :

```
!300:PHA
! LDA #'Z
! JSR $FDED
! PLA
! RTS
```

Vous voyez :

```
0300- 48          PHA
0301- A9 DA       LDA #$DA
0303- 20 ED FD    JSR $FDED
0306- 68          PLA
0307- 60          RTS
```

Notez que l'on ne peut pas entrer des commandes du moniteur lorsque l'on est dans le mini-assembleur.
On revient au moniteur en tapant <Return> à vide.

## Le Basic Applesoft

Celui-ci accepte instructions et variables en minuscules, qui seront automatiquement converties en majuscules. Saluons cette amélioration qui fait que, sous ProDOS, on peut entrer n'importe quelle commande (SED, Basic, Moniteur) indifféremment en majuscules ou en minuscules. Finie la gymastique du "CAPS LOCK" ! Cela, ajouté à l'amélioration du clavier, apporte un grand confort d'utilisation...

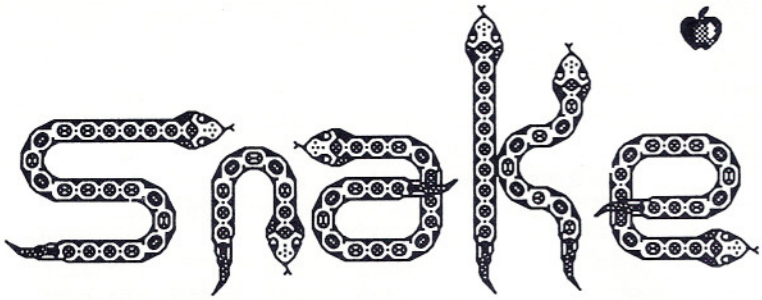
Mis à part cela, les routines "tabulantes" du Basic Applesoft fonctionnent maintenant parfaitement en 80 colonnes. Il s'agit de HTAB (on peut enfin écrire "HTAB 60" sans provoquer un résultat aléatoire !), TAB, SPC et la virgule de tabulation dans les PRINT.

Remarquons qu'aucun effort n'a été fait pour rajouter des instructions, ou pour en enrichir d'autres (par exemple pour l'utilisation de la double résolution), et que les "bugs" célèbres sont même restés, tels celui de la virgule de tabulation (cela aurait requis le changement d'un seul octet), ou celui du GOTO/GOSUB 440xxx...

## Pascal

Outre les fonctions CTRL-F et CTRL-E décrites plus haut, et le rejet du code permettant de passer en 40 colonnes sur l'Apple Pascal 1.1, la grande nouveauté est que la nouvelle ROM moniteur peut démarrer le Pascal depuis un autre périphérique qu'un DISK II dans le slot 6. Il est donc désormais possible de démarrer depuis un UNIDISK II, un PROFILE, etc.

Nous avons fait le tour des spécificités de la nouvelle ROM qui, malgré de nombreuses lacunes, réprésente néanmoins un net progrès par rapport à l'ancienne.



*Philippe Krepper*

Ce petit jeu d'adresse peut tourner sur n'importe quel type d'Apple : il s'agit, avec un serpent, de 'manger' des pommes représentées par un petit carré sur l'écran. Suivant le nombre d'erreurs déjà commises, chaque prise rapporte de un à trois points.

Si le serpent heurte le bord de l'écran, ou se touche lui-même, il commet une erreur. Au-delà de trois erreurs, il est disqualifié (et vous avec !).

Pour déplacer le serpent, utilisez les flèches droite et gauche. Attention, il faut raisonner comme si vous vous trouviez à la place du serpent et donc le diriger en conséquence.

Enfin, pour corser l'histoire, le serpent s'agrandit à chaque prise...

## *Quelques notes :*

*Le programme se sert de l'espace $800 à $BFF pour stocker une table d'adresses HGR qu'il crée lui-même.*

*Le score du jeu est stocker sur la disquette en piste $11, secteur $0, octet 4 et 5, emplacement inutilisé (cf. manuel DOS).*

*Pour éviter les problèmes avec d'autres programmes qui utiliseraient cette adresse, laissez ce jeu seul sur sa disquette.*

## *Comment faire ?*

***Vous disposez** de la disquette d'accompagnement, faites simplement 'BRUN SNAKE'.*

***Vous ne disposez pas** de cette disquette, alors vous devrez :*

1. *Saisir et sauvegarder SNAKE.OBJ.*
   *(C'est le programme objet créé par le source SNAKE.TXT).*
2. *Saisir et sauvegarder SNAKE.MASQUE.*
   *Il s'agit de la représentation mémoire d'un écran texte. Bien sûr on ne peut le saisir directement à l'adresse $400, vous le saisirez à partir de l'adresse $1000.*
3. *Saisir et sauvegarder SNAKE.FUSION.*
   *Ce programme charge SNAKE.OBJ, SNAKE.MASQUE dans la mémoire écran et sauvegarde le tout sous la forme d'un fichier exécutable nommé SNAKE*
4. *Pour toutes les utilisation ultérieures, il suffira de taper 'BRUN SNAKE'*

## Source 'SNAKE.TXT'

### Assembleur LISA 1.5

```
1          ORG $C00
2  LGSNAK EPZ $6
3  SENS   EPZ $7
4  VISUAL EPZ $8
5  POMMEL EPZ $9
6  POMMEC EPZ $A
7  DELAI  EPZ $B
8  SCORE  EPZ $18
9  HTAB   EPZ $24
10 LIGNE  EPZ $26
11 SNAKES EPZ $CE
12 TEMP0  EPZ $D0
13 TEMP1  EPZ $D1
14 TEMP2  EPZ $D2
15 TEMP3  EPZ $D3
16 TEMP4  EPZ $D4
17 RECORD EPZ $D6
18 JPRWTS EQU $3D9
19 HGRBAS EQU $800
20 HGRHAU EQU $900
21 MASQUE EQU $A00
22 XD7    EQU $B00
23 VOLUME EQU $B7EB
24 PISTE  EQU $B7EC
25 SECTEU EQU $B7ED
26 CMD    EQU $B7F4
27 BUFFER EQU $B7F0
28 TOUCHE EQU $C000
29 RAZKBD EQU $C010
30 SPEAKR EQU $C030
31 GRAPHI EQU $C050
32 TEXTE  EQU $C051
33 NONMXT EQU $C052
34 MIXTE  EQU $C053
35 HAURES EQU $C057
36 LINPRT EQU $ED24
37 HGR    EQU $F3E2
38 HPOSN  EQU $F411
39 CLOCHE EQU $FBE4
40 PAUSE  EQU $FCA8
41 CHROUT EQU $FDED
42 ; INITIALISATION
43         JSR HGR
44         LDA TEXTE
45         LDA #$15
46         STA HTAB+1
47         JSR $FC22
48         LDA NONMXT
49         LDY #0
50         STY VISUAL
51         STY TEMP0
52         STY COLOR
53 ;
54         LDA #1
55         STA CMD
56         LDA #0
57         STA SECTEU
58         STA VOLUME
59         LDA #$11
60         STA PISTE
61         LDA #SECTOR
62         STA BUFFER
63         LDA /SECTOR
64         STA BUFFER+1
65         LDA #$B7
66         LDY #$E8
67         JSR JPRWTS
68         BCC NOTERR
69         JMP $3D0
70 NOTERR LDA SECTOR+$4
```

```
 71          STA RECORD
 72          LDA SECTOR+$5
 73          STA RECORD+1
 74  ;
 75          LDA #RESET
 76          STA $3F2
 77          LDA /RESET
 78          STA $3F3
 79          EOR #$A5
 80          STA $3F4
 81  ;CREATION TABLE
   MASQUE ET XD7
 82  INIT1   LDX TEMP0
 83          LDA #0
 84          TAY
 85          JSR HPOSN
 86          LDX TEMP0
 87          LDA $E5
 88          STA XD7,X
 89          LDA $30
 90          STA MASQUE,X
 91          DEC TEMP0
 92          BNE INIT1
 93  ;CREATION TABLE
   HGRBAS ET HGRHAU
 94  INIT2   LDA TEMP0
 95          CMP #$C0
 96          BCS SUITE
 97          LDY #0
 98          JSR HPOSN
 99          LDY TEMP0
100          LDA LIGNE
101          STA HGRBAS,Y
102          LDA LIGNE+1
103          STA HGRHAU,Y
104          INC TEMP0
105          BNE INIT2
106  ;RAZ DU TABLEAU
   ECRAN
107  SUITE   LDA #0
108          STA TEMP3
109          STA TEMP4
110  INIT3   LDX TEMP3
111          CPX #$30
112          BCS NEXT
113          LDY TEMP4
114          JSR CARRE
115          INC TEMP3
116          BNE INIT3
117  NEXT    LDA #0
118          STA TEMP3
119          INC TEMP4
120          LDA TEMP4
121          CMP #$40
122          BCC INIT3
123  ;TRACÉ DU CADRE DE
   L'ECRAN
124          LDA #0
125          STA TEMP3
126          DEC COLOR
127  CADRE1  LDY TEMP3
128          CPY #$40
129          BCS VRTCAL
130          LDX #0
131          JSR CARRE
132          LDY TEMP3
133          LDX #$2F
134          JSR CARRE
135          INC TEMP3
136          BNE CADRE1
137  VRTCAL  LDA #$2E
138          STA TEMP3
139  CADRE2  LDX TEMP3
140          LDY #0
141          JSR CARRE
142          LDX TEMP3
143          LDY #$3F
144          JSR CARRE
145          DEC TEMP3
146          BNE CADRE2
147  ;
148  DEBUT   LDA #2
149  ;LONGUEUR DU SNAKE
150          STA LGSNAK
151          STA TEMP3
152  BCL01   LDX TEMP3
153          LDA #$10
154          STA SNAKEC,X
155          LDA TEMP3
156          STA SNAKEL,X
157          TAX
158          LDY #$10
159          JSR CARRE
160          DEC TEMP3
161          BNE BCL01
162          JSR PLACE
163          LDA #3
164          STA SNAKES
165          LDX #0
166          STX HTAB
167          STX DELAI
168          STX SCORE
169          STX SCORE+1
170          INX
171          STX SENS
172          JSR KPRESS
173          LDA GRAPHI
174          LDY #$23
175  MESS    LDA MESSAG,Y
176          JSR CHROUT
177          DEY
178          BPL MESS
179          LDA #$24
180          STA HTAB
181          LDX RECORD
182          LDA RECORD+1
183          JSR LINPRT
184  ;BOUCLE D'UNE
   PARTIE
185  JEU     LDX SNAKEL+1
186          STX TEMP3
187          LDY SNAKEC+1
188          STY TEMP4
189          LDY SENS
190          DEY
191          BEQ SENS1
192          DEY
193          BEQ SENS2
194          DEY
195          BEQ SENS3
196  SENS4   DEC TEMP3
197          JMP TSTPOM
198  SENS3   INC TEMP4
199          JMP TSTPOM
200  SENS2   INC TEMP3
201          JMP TSTPOM
202  SENS1   DEC TEMP4
203  TSTPOM  LDX TEMP3
204          LDY TEMP4
205          CPX POMMEL
206          BNE TSTVID
207          CPY POMMEC
208          BNE TSTVID
209  MANGE   INC LGSNAK
210          JSR ALLONGE
211          JSR PLACE
212          DEC DELAI
213          LDA #$15
214          STA HTAB
215          CLC
216          LDA SCORE
217          ADC SNAKES
218          STA SCORE
219          BCC LDXSCO
220          INC SCORE+1
221  LDXSCO  LDX SCORE
222          LDA SCORE+1
223          JSR LINPRT
224          LDY #0
225  BIP1    TYA
226          TAX
227  BIP2    LDA SPEAKR
228          DEX
229          BNE BIP2
230          LDX #$30
231  BIP3    DEX
232          BNE BIP3
233          INY
234          CPY #$80
235          BNE BIP1
236  BIP4    LDX #$10
237  BIP5    LDA SPEAKR
238          DEX
239          BNE BIP5
240          TYA
241          TAX
242  BIP6    DEX
243          BNE BIP6
244          DEY
245          BNE BIP4
246          JMP FINTST+3
247  TSTVID  JSR CLCADR
248          LDA (TEMP0),Y
249          BNE CRASH
250          JMP FINTST
251  ; PLANTE
252  CRASH   DEC SNAKES
253          LDA SNAKES
254          ADC #7
255          STA HTAB
256          LDA #" "
257          JSR CHROUT
258          LDA SNAKES
259          BEQ FIN
260          LDX #0
261  BIP7    TXA
262          TAY
263  BIP8    LDA SPEAKR
264          DEY
265          BNE BIP8
266          TXA
267          TAY
268  BIP9    DEY
269          BNE BIP9
270          LDY #$20
271  BIP10   LDA SPEAKR
272          DEY
273          BNE BIP10
274          LDY #$20
275  BIP11   DEY
276          BNE BIP11
277          INX
278          BNE BIP7
279          JSR KPRESS
280          BMI FINTST+3
281  ;FIN D'UNE PARTIE
282  FIN     LDY #$40
283  BIP13   STY TEMP3
284  BIP14   TYA
285          TAX
286  BIP15   LDA SPEAKR
287          DEX
288          BNE BIP15
289          LDX TEMP3
290  BIP16   DEX
291          BNE BIP16
292          DEC TEMP3
293          BNE BIP14
294          DEY
295          BNE BIP13
296          LDA SCORE+1
297          CMP RECORD+1
298          BCC PAS.>
299          BNE SCORE>
300          LDA SCORE
301          CMP RECORD
302          BCC PAS.>
303  SCORE>  LDA #$24
304          STA HTAB
305          LDX SCORE
306          STX RECORD
307          LDA SCORE+1
308          STA RECORD+1
309          JSR LINPRT
310          LDX #3
311  BIP12   LDA #$80
312          JSR PAUSE
313          LDY #$FF
314          JSR CLOCHE
315          DEX
316          BPL BIP12
317  PAS.>   LDA TEXTE
318          INC COLOR
319          LDX POMMEL
320          LDY POMMEC
321          JSR CARRE
322          LDY LGSNAK
323          STY TEMP3
324  EFFACE  LDY TEMP3
325          LDA SNAKEL,Y
```

```
326          TAX
327          LDA SNAKEC,Y
328          TAY
329          JSR CARRE
330          DEC TEMP3
331          BNE EFFACE
332          DEC COLOR
333          JMP DEBUT
334  ;
335  FINTST  JSR DECALE
336          LDA DELAI
337          STA TEMP3
338  BCL04   JSR LITSNS
339          JSR LITSNS
340          JSR LITSNS
341          JSR LITSNS
342          JSR LITSNS
343          JSR LITSNS
344          JSR LITSNS
345          JSR LITSNS
346          JSR LITSNS
347          DEC TEMP3
348          BNE BCL04
349          JMP JEU
350  ;PLACE UNE POMME
351  PLACE   ADC $C060
352          ADC $C00,X
353          TAX
354          CPX #3
355          BCC PLACE
356          CPX #$2D
357          BCS PLACE
358          ADC $C060
359          ADC $D00,Y
360          TAY
361          CPX #3
362          BCC PLACE
363          CPY #$3D
364          BCS PLACE
365          JSR CLCADR
366          LDA (TEMP0),Y
367          BNE PLACE
368          STX POMMEL
369          STY POMMEC
370          BEQ CARRE
371  ; DECALE LE SNAK
372  DECALE  LDY LGSNAK
373          LDA SNAKEL,Y
374          TAX
375          LDA SNAKEC,Y
376          TAY
377          INC COLOR
378          JSR CARRE
379          DEC COLOR
380  ALLONG  LDY LGSNAK
381  BCL03   LDA SNAKEL-1,Y
382          STA SNAKEL,Y
383          LDA SNAKEC-1,Y
384          STA SNAKEC,Y
385          DEY
386          BNE BCL03
387          LDX TEMP3
388          STX SNAKEL+1
389          LDY TEMP4
390          STY SNAKEC+1
391          JSR CARRE
392          RTS
393  ;AFFICHE/EFFACE UN
       CARRE EN LIGNE X
       COLONE Y.
394  CARRE   JSR CLCADR
395          LDA COLOR
396          STA (TEMP0),Y
397          TXA
398          ASL
399          ASL
400          STA TEMP0
401          TYA
402          ASL
403          ASL
404          STA TEMP1
405          LDA #4
406          STA TEMP2
407  CARRE1  LDY TEMP0
408          LDX TEMP1
409          JSR PLOT
410          INX
411          LDY TEMP0
412          JSR PLOTML
413          INX
414          LDY TEMP0
415          JSR PLOTML
416          INX
417          LDY TEMP0
418          JSR PLOTML
419          INC TEMP0
420          DEC TEMP2
421          BNE CARRE1
422          RTS
423  ;CALCUL L'ADRESSE
       D'UNE LIGNE X
       ET LA PLACE DANS
       TEMP0 , TEMP1
424  CLCADR  STX TEMP0
425          LDA #0
426          STA TEMP1
427          ASL TEMP0
428          ROL TEMP1
429          ASL TEMP0
430          ROL TEMP1
431          ASL TEMP0
432          ROL TEMP1
433          ASL TEMP0
434          ROL TEMP1
435          ASL TEMP0
436          ROL TEMP1
437          ASL TEMP0
438          ROL TEMP1
439          LDA #ECRAN
440          ADC TEMP0
441          STA TEMP0
442          LDA /ECRAN
443          ADC TEMP1
444          STA TEMP1
445          RTS
446  ;BIP...BIP...
447  KPRESS  LDA RAZKBD
448  REPEAT  BIT TOUCHE
449          BMI RETOUR
450          LDY #9
451          JSR CLOCHE
452          LDA #0
453          JSR PAUSE
454          JSR PAUSE
455          BEQ REPEAT
456  ;BASCULE DE
       L'AFFICHAGE DU
       SCORE SI ESC
457  LITSNS  BIT TOUCHE
458          BPL RETOUR
459          LDA TOUCHE
460          BIT RAZKBD
461          CMP #$88
462          BEQ DROITE
463          CMP #$95
464          BEQ GAUCHE
465          CMP #$8D
466          BEQ BASCUL
467          CMP #$9B
468          BNE RETOUR
469  STOP    JSR KPRESS
470          BMI LITSNS
471  BASCUL  LDA #$FF
472          STA TOUCHE
473          EOR VISUAL
474          STA VISUAL
475          BEQ CACHE
476          LDA MIXTE
477  RETOUR  RTS
478  CACHE   LDA NONMXT
479          RTS
480  DROITE  LDY SENS
481          INY
482          CPY #5
483          BCC STKSNS
484          DEY
485          DEY
486          DEY
487          DEY
488  STKSNS  STY SENS
489          RTS
490  GAUCHE  LDY SENS
491          DEY
492          BNE STKSNS
493          INY
494          INY
495          INY
496          INY
497          BNE STKSNS
498  ;ALLUME/ETEIND UN
       POINT LIGNE Y
       COLONE X
499  PLOT    LDA HGRBAS,Y
500          STA LIGNE
501          LDA HGRHAU,Y
502          STA LIGNE+1
503  PLOTML  LDA XD7,X
504          TAY
505          HEX A9
506  COLOR   HEX 00
507          EOR (LIGNE),Y
508          AND MASQUE,X
509          EOR (LIGNE),Y
510          STA (LIGNE),Y
511          RTS
512  RESET   LDA #2
513          STA CMD
514          LDA RECORD
515          STA SECTOR+$4
516          LDA RECORD+1
517          STA SECTOR+$5
518          LDA #$B7
519          LDY #$E8
520          JSR JPRWTS
521          LDA #$6
522          JMP $FE95
523  MESSAG  ASC ":DROCER
       /     :EROCS  /
       ààà:SEKANS"
524  SNAKEL  DFS $100
525  SNAKEC  DFS $100
526  SECTOR  DFS $100
527  ECRAN   END
```

## Programme 'SNAKE.FUSION'

```
10  REM CHARGE SNAKE.OBJ ET SNAKE MASQUE DANS LA MEMOIRE ECRAN ET SAUVEGARDE LE TOUT SOUS LA
    FORME D'UN FICHIER EXECUTABLE NOMME SNAKE
20 M$ = "SNAKE.MASQUE"
30 F$ = "SNAKE"
40  POKE 973,76: POKE 974,00: POKE 975,12
50 D$ =  CHR$ (4): PRINT D$"BLOAD "M$",A$400": PRINT D$"BLOAD SNAKE.OBJ" : PRINT D$"BSAVE "F$
    ",A973,L3111"
```

## Fichier 'SNAKE.MASQUE'

*Après avoir saisi ce code sous moniteur, vous le sauvegarderez par BSAVE SNAKE.MASQUE,A$1000,L$400*

```
1000- AD AD AD AD AD AD AD AD
1008- AD AD AD AD AD AD AD AD
1010- AD AD AD AD AD AD AD AD
1018- AD AD AD AD AD AD AD AD
1020- AD AD AD AD AD AD AD AD
1028- A0 A0 A0 A0 A0 A0 A0 A0
1030- A0 A0 A0 A0 A0 A0 A0 A0
1038- A0 A0 A0 A0 A0 A0 A0 A0
1040- A0 A0 A0 A0 A0 A0 A0 A0
1048- A0 A0 A0 A0 A0 A0 A0 A0
1050- A0 A0 A0 A0 A0 A0 A0 A0
1058- A0 A0 A0 A0 A0 A0 A0 A0
1060- A0 A0 A0 A0 A0 A0 A0 A0
1068- A0 A0 A0 A0 A0 A0 A0 A0
1070- A0 A0 A0 A0 A0 A0 A0 A0
1078- 13 A0 A0 A0 A0 A0 26 A0
1080- 20 20 20 20 20 20 20 20
1088- 20 20 20 2A 2A 2A 20 13
1090- 20 0E 20 01 20 0B 20 05
1098- 20 2A 2A 2A 20 20 20 20
10A0- 20 20 20 20 20 20 20 20
10A8- A0 A0 D4 CF D5 C3 C8 C5
10B0- D3 BA A0 A0 A0 2D 3E A0
10B8- D4 CF D5 D2 CE C5 A0 C1
10C0- A0 C4 D2 CF C9 D4 C5 AE
10C8- A0 A0 A0 A0 A0 A0 A0 A0
10D0- A0 A0 A0 A0 A0 A0 A0 A0
10D8- A0 A0 A0 A0 A0 A0 A0 A0
10E0- A0 A0 A0 A0 A0 A0 A0 A0
10E8- A0 A0 A0 A0 A0 A0 A0 A0
10F0- A0 A0 A0 A0 A0 A0 A0 A0
10F8- 04 FF FF FF FF FF 36 FF
1100- AD AD AD AD AD AD AD AD
1108- AD AD AD AD AD AD AD AD
1110- AD AD AD AD AD AD AD AD
1118- AD AD AD AD AD AD AD AD
1120- AD AD AD AD AD AD AD AD
1128- A0 A0 AD AD AD AD AD AD
1130- AD A0 A0 A0 A0 3C 2D A0
1138- D4 CF D5 D2 CE C5 A0 C1
1140- A0 C7 C1 D5 C3 C8 C5 AE
1148- A0 A0 A0 A0 A0 A0 A0 A0
1150- A0 C1 D0 D0 D5 D9 C5 DA
1158- A0 D3 D5 D2 A0 D5 CE 85
1160- A0 D4 CF D5 C3 C8 C5 A0
1168- A0 D0 CF D5 D2 A0 C3 CF
1170- CD CD C5 CE C3 C5 D2 A0
1178- 26 27 27 27 27 27 27 27
1180- A0 A0 A0 A0 A0 A0 A0 A0
1188- A0 A0 A0 A0 A0 A0 A0 A0
1190- A0 A0 A0 A0 A0 A0 A0 A0
1198- A0 A0 A0 A0 A0 A0 A0 A0
11A0- A0 A0 A0 A0 A0 A0 A0 A0
11A8- A0 A0 A0 A0 A0 A0 A0 A0
11B0- A0 A0 A0 A0 A0 A0 A0 A0
11B8- A0 A0 A0 A0 A0 A0 A0 A0
11C0- A0 A0 A0 A0 A0 A0 A0 A0
11C8- A0 A0 A0 A0 A0 A0 A0 A0
11D0- A0 A0 A0 A0 A0 A0 A0 A0
11D8- A0 A0 A0 A0 A0 A0 A0 A0
11E0- A0 A0 A0 A0 A0 A0 A0 A0
11E8- A0 A0 A0 A0 A0 A0 A0 A0
11F0- A0 A0 A0 A0 A0 A0 A0 A0
11F8- 60 FF FF FF FF FF FF FF
1200- A0 A0 A0 A0 A0 A0 A0 A0
1208- A0 A0 A0 A0 A0 A0 A0 A0
1210- A0 A0 A0 A0 A0 A0 A0 A0
1218- A0 A0 A0 A0 A0 A0 CB D2
1220- C5 D0 D0 C5 D2 A0 D0 C8
1228- A0 A0 A0 A0 A0 A0 A0 A0
1230- A0 A0 A0 A0 05 13 03 A0
1238- D0 C1 D5 D3 C5 AE A0 A0
1240- A0 A0 A0 A0 A0 A0 A0 A0
1248- A0 A0 A0 A0 A0 A0 A0 A0
1250- AD AD AD AD AD AD AD AD
1258- AD AD AD AD AD AD AD AD
1260- AD AD AD AD AD AD AD AD
1268- AD AD AD AD AD AD AD AD
1270- AD AD AD AD AD AD AD AD
1278- 60 27 27 27 27 27 27 27
1280- A0 A0 A0 A0 A0 A0 A0 A0
1288- A0 A0 A0 A0 A0 A0 A0 A0
1290- A0 A0 A0 A0 A0 A0 A0 A0
1298- A0 A0 A0 A0 A0 A0 CA D5
12A0- C9 CC CC C5 D4 A0 B8 B5
12A8- A0 A0 A0 A0 A0 A0 A0 A0
12B0- A0 A0 A0 A0 A0 A0 A0 A0
12B8- A0 A0 A0 A0 A0 A0 A0 A0
12C0- A0 A0 A0 A0 A0 A0 A0 A0
12C8- A0 A0 A0 A0 A0 A0 A0 A0
12D0- A0 A0 A0 A0 A0 A0 A0 A0
12D8- A0 A0 A0 A0 A0 A0 A0 A0
12E0- A0 A0 A0 A0 A0 A0 A0 A0
12E8- A0 A0 A0 A0 A0 A0 A0 A0
12F0- A0 A0 A0 A0 A0 A0 A0 A0
12F8- 02 FF FF FF FF FF FF FF
1300- A0 A0 A0 A0 A0 A0 A0 A0
1308- A0 A0 A0 A0 A0 A0 A0 A0
1310- A0 A0 A0 A0 A0 A0 A0 A0
1318- A0 A0 A0 A0 A0 A0 A0 A0
1320- A0 A0 A0 A0 A0 A0 A0 A0
1328- A0 A0 A0 A0 A0 A0 A0 A0
1330- A0 12 05 14 15 12 0E A0
1338- CF CE AF CF C6 C6 A0 C1
1340- C6 C6 C9 C3 C8 C1 C7 C5
1348- A0 A0 A0 A0 A0 A0 A0 A0
1350- AD AD AD AD AD AD AD AD
1358- AD AD AD AD AD AD AD AD
1360- AD AD AD AD AD AD AD AD
1368- AD AD AD AD AD AD AD AD
1370- AD AD AD AD AD AD AD AD
1378- 27 27 27 27 27 27 27 27
1380- A0 A0 A0 A0 A0 A0 A0 A0
1388- A0 A0 A0 A0 A0 A0 A0 A0
1390- A0 A0 A0 A0 A0 A0 A0 A0
1398- A0 A0 A0 A0 A0 A0 A0 A0
13A0- A0 A0 A0 A0 A0 A0 A0 A0
13A8- A0 A0 A0 A0 A0 A0 A0 A0
13B0- A0 A0 A0 A0 A0 A0 A0 A0
13B8- C4 D5 A0 D3 C3 CF D2 C5
13C0- AE A0 A0 A0 A0 A0 A0 A0
13C8- A0 A0 A0 A0 A0 A0 A0 A0
13D0- A0 A0 A0 A0 A0 A0 A0 A0
13D8- A0 A0 A0 A0 A0 A0 A0 A0
13E0- A0 A0 A0 A0 A0 A0 A0 A0
13E8- A0 A0 A0 A0 A0 A0 A0 A0
13F0- A0 A0 A0 A0 A0 A0 A0 A0
13F8- C1 FF FF FF FF FF FF 00
```

## Récapitulation 'SNAKE.OBJ'

*Après avoir saisi ce code sous moniteur, vous le sauvegarderez par BSAVE SNAKE.OBJ,A$C00,L$3F4*

```
0C00- 20 E2 F3 AD 51 C0 A9 15
0C08- 85 25 20 22 FC AD 52 C0
0C10- A0 00 84 08 84 D0 8C AA
0C18- 0F A9 01 8D F4 B7 A9 00
0C20- 8D ED B7 8D EB B7 A9 11
0C28- 8D EC B7 A9 F4 8D F0 B7
0C30- A9 11 8D F1 B7 A9 B7 A0
0C38- E8 20 D9 03 90 03 4C D0
0C40- 03 AD F8 11 85 D6 AD F9
0C48- 11 85 D7 A9 B5 8D F2 03
0C50- A9 0F 8D F3 03 49 A5 8D
0C58- F4 03 A6 D0 A9 00 A8 20
0C60- 11 F4 A6 D0 A5 E5 9D 00
0C68- 0B A5 30 9D 00 0A C6 D0
0C70- D0 E8 A5 D0 C9 C0 B0 15
0C78- A0 00 20 11 F4 A4 D0 A5
0C80- 26 99 00 08 A5 27 99 00
0C88- 09 E6 D0 D0 E5 A9 00 85
0C90- D3 85 D4 A6 D3 E0 30 B0
0C98- 09 A4 D4 20 D7 0E E6 D3
0CA0- D0 F1 A9 00 85 D3 E6 D4
0CA8- A5 D4 C9 40 90 E5 A9 00
0CB0- 85 D3 CE AA 0F A4 D3 C0
0CB8- 40 B0 10 A2 00 20 D7 0E
0CC0- A4 D3 A2 2F 20 D7 0E E6
0CC8- D3 D0 EA A9 2E 85 D3 A6
0CD0- D3 A0 00 20 D7 0E A6 D3
0CD8- A0 3F 20 D7 0E C6 D3 D0
0CE0- EE A9 02 85 06 85 D3 A6
0CE8- D3 A9 10 9D F4 10 A5 D3
0CF0- 9D F4 0F AA A0 10 20 D7
0CF8- 0E C6 D3 D0 EA 20 7A 0E
0D00- A9 03 85 CE A2 00 86 24
0D08- 86 0B 86 18 86 19 E8 86
0D10- 07 20 38 0F AD 50 C0 A0
0D18- 23 B9 D0 0F 20 ED FD 88
0D20- 10 F7 A9 24 85 24 A6 D6
0D28- A5 D7 20 24 ED AE F5 0F
0D30- 86 D3 AC F5 10 84 D4 A4
0D38- 07 88 F0 15 88 F0 0D 88
0D40- F0 05 C6 D3 4C 53 0D E6
0D48- D4 4C 53 0D E6 D3 4C 53
0D50- 0D C6 D4 A6 D3 A4 D4 E4
0D58- 09 D0 4B C4 0A D0 47 E6
0D60- 06 20 B8 0E 20 7A 0E C6
0D68- 0B A9 15 85 24 18 A5 18
0D70- 65 CE 85 18 90 02 E6 19
0D78- A6 18 A5 19 20 24 ED A0
0D80- 00 98 AA AD 30 C0 CA D0
0D88- FA A2 30 CA D0 FD C8 C0
0D90- 80 D0 EE A2 10 AD 30 C0
0D98- CA D0 FA 98 AA CA D0 FD
0DA0- 88 D0 F0 4C 54 0E 20 0D
0DA8- 0F B1 D0 D0 03 4C 51 0E
0DB0- C6 CE A5 CE 69 07 85 24
0DB8- A9 A0 20 ED FD A5 CE F0
0DC0- 24 A2 00 8A A8 AD 30 C0
0DC8- 88 D0 FA 8A A8 88 D0 FD
0DD0- A0 20 AD 30 C0 88 D0 FA
0DD8- A0 20 88 D0 FD E8 D0 E3
0DE0- 20 38 0F 30 6F A0 40 84
0DE8- D3 98 AA AD 30 C0 CA D0
0DF0- FA A6 D3 CA D0 FD C6 D3
0DF8- D0 EF 88 D0 EA A5 19 C5
0E00- D7 90 26 D0 06 A5 18 C5
0E08- D6 90 1E A9 24 85 24 A6
0E10- 18 86 D6 A5 19 85 D7 20
0E18- 24 ED A2 03 A9 80 20 A8
0E20- FC A0 FF 20 E4 FB CA 10
0E28- F3 AD 51 C0 EE AA 0F A6
0E30- 09 A4 0A 20 D7 0E A4 06
0E38- 84 D3 A4 D3 B9 F4 0F AA
0E40- B9 F4 10 A8 20 D7 0E C6
0E48- D3 D0 EF CE AA 0F 4C E1
0E50- 0C 20 A5 0E A5 0B 85 D3
0E58- 20 4F 0F 20 4F 0F 20 4F
0E60- 0F 20 4F 0F 20 4F 0F 20
0E68- 4F 0F 20 4F 0F 20 4F 0F
0E70- 20 4F 0F C6 D3 D0 E1 4C
0E78- 2D 0D 6D 60 C0 7D 00 0C
0E80- AA E0 03 90 F5 E0 2D B0
0E88- F1 6D 60 C0 79 00 0D A8
0E90- E0 03 90 E6 C0 3D B0 E2
0E98- 20 0D 0F B1 D0 D0 DB 86
0EA0- 09 84 0A F0 32 A4 06 B9
0EA8- F4 0F AA B9 F4 10 A8 EE
0EB0- AA 0F 20 D7 0E CE AA 0F
0EB8- A4 06 B9 F3 0F 99 F4 0F
0EC0- B9 F3 10 99 F4 10 88 D0
0EC8- F1 A6 D3 8E F5 0F A4 D4
0ED0- 8C F5 10 20 D7 0E 60 20
0ED8- 0D 0F AD AA 0F 91 D0 8A
0EE0- 0A 0A 85 D0 98 0A 0A 85
0EE8- D1 A9 04 85 D2 A4 D0 A6
0EF0- D1 20 9B 0F E8 A4 D0 20
0EF8- A5 0F E8 A4 D0 20 A5 0F
0F00- E8 A4 D0 20 A5 0F E6 D0
0F08- C6 D2 D0 E1 60 86 D0 A9
0F10- 00 85 D1 06 D0 26 D1 06
0F18- D0 26 D1 06 D0 26 D1 06
0F20- D0 26 D1 06 D0 26 D1 06
0F28- D0 26 D1 A9 F4 65 D0 85
0F30- D0 A9 12 65 D1 85 D1 60
0F38- AD 10 C0 2C 00 C0 30 3D
0F40- A0 09 20 E4 FB A9 00 20
0F48- A8 FC 20 A8 FC F0 EC 2C
0F50- 00 C0 10 29 AD 00 C0 2C
0F58- 10 C0 C9 88 F0 24 C9 95
0F60- F0 2E C9 8D F0 09 C9 9B
0F68- D0 13 20 38 0F 30 E0 A9
0F70- FF 8D 00 C0 45 08 85 08
0F78- F0 04 AD 53 C0 60 AD 52
0F80- C0 60 A4 07 C8 C0 05 90
0F88- 04 88 88 88 88 84 07 60
0F90- A4 07 88 D0 F8 C8 C8 C8
0F98- C8 D0 F2 B9 00 08 85 26
0FA0- B9 00 09 85 27 BD 00 0B
0FA8- A8 A9 00 51 26 3D 00 0A
0FB0- 51 26 91 26 60 A9 02 8D
0FB8- F4 B7 A5 D6 8D F8 11 A5
0FC0- D7 8D F9 11 A9 B7 A0 E8
0FC8- 20 D9 03 A9 06 4C 95 FE
0FD0- BA C4 D2 CF C3 C5 D2 A0
0FD8- A0 AF A0 A0 A0 A0 A0 BA
0FE0- C5 D2 CF C3 D3 A0 A0 AF
0FE8- A0 A0 C0 C0 C0 BA D3 C5
0FF0- CB C1 CE D3
```

# Micro-informations

## Jean-Michel Gourévitch

Apple serait-il donc devenu un constructeur de micro-ordinateurs comme les autres ? Il semble en tous cas bien loin le temps où Steve Jobs annonçait une nouvelle race de micro-ordinateurs «for the rest of us». Andy Hertzfeld, un des développeurs du Macintosh et le père du numériseur Thunderscan, l'exprimait récemment ainsi : «Du temps de Jobs, la mission d'Apple était de rendre les ordinateurs accessibles, maintenant, c'est de protéger ses actionnaires».

Aujourd'hui, les fabricants des ordinateurs à la pomme emploient les mêmes procédés (discutables) que tous les autres constructeurs. En commençant par annoncer des produits indisponibles. Annoncée en janvier, la transformation des Macintosh en Macintosh Plus n'avait, en mai, été réalisée que pour une toute petite minorité d'heureux veinards. Chez les concessionnaires, lorsqu'on abordait le sujet, les mines s'allongeaient... presque autant que les listes d'attente. Les délais ? Juillet, août ? Certes, le Macintosh est un succès et il faut s'en réjouir il est vrai, qu'il est plus rentable pour Apple de vendre des Macintosh Plus entiers que des "kits" de mise à niveau. Il n'empêche : les candidats à la transformation l'ont trouvé plutôt saumâtre. Au moins, ont-ils été ainsi épargnés par les problèmes de compatibilité qui ont affecté les possesseurs de Macintosh Plus. A cause d'abord d'un vilain "bug" dans le Système (version 3.1). Apple a du, en toute hâte, livrer une nouvelle version (la 3.1.1). En raison, ensuite, du fonctionnement même du nouveau système de hiérarchisation des fichiers. Les gourous d'Apple avaient pourtant juré *urbi et orbi* leurs grands dieux que tous les logiciels "bien élevés" devaient fonctionner sans bavures sur le Macintosh Plus. Las ! Il n'en est apparemment rien. Pom's a relevé, dans ce numéro, de nombreux exemples, voici donc juste quelques conseils supplémentaires :

- aux utilisateurs d'Excel : garder tous les documents liés dans le même dossier, pour que les mises à jour se fassent bien automatiquement ;
- aux utilisateurs de MacDraw : ne vous étonnez pas de ne pouvoir afficher plus de 11 polices de caractères. Pas question de faire défiler ce menu ;
- aux utilisateurs du Switcher : ajouter 20Ko par application et placez les toutes dans le même dossier.

Problèmes de jeunesse, certes, mais qui pourraient bien dégoûter à jamais ceux qui ont commencé leur parcours micro-informatique sur le Macintosh Plus. Dernier exemple du peu de cas que semble faire Apple de ses clients : l'affaire du nouveau Macintosh 512Ko. Ce dernier bébé de la gamme Macintosh est simplement un Mac de 512Ko doté d'un lecteur interne de disquettes double face (800Ko contre 400 pour l'ancien) et de la nouvelle ROM de 128Ko. Par rapport au Macintosh Plus, il lui manque donc la mémoire de 1 Méga et l'interface SCSI pour disque dur. Aux États-Unis, il a été annoncé au prix de 1995 dollars. Une bonne occasion, au cours du dollar, pour permettre aux moins riches de s'équiper d'un Mac à 15 000,00 F ? Eh bien non, Apple France n'entend pas le vendre moins de 20 000,00 F. Cette fois c'est pour préserver les marges des revendeurs. Question au PDG d'Apple : et le client alors ? Pourquoi ne pas refaire le même bon geste effectué en tarifant avantageusement chez nous les mises à niveau du Mac en Macintosh Plus ?

Moyennant quoi, tout va très bien pour Apple. Bien qu'elle ait diminué son chiffre d'affaires, la firme a augmenté aux Etats Unis son bénéfice de 218%. La revue Infoworld, qui fait autorité dans le monde de la micro a salué l'arrivée du Macintosh Plus en le dotant d'une note de 7,7, en le "recommandant hautement", et en estimant qu'Apple pouvait désormais attaquer de pied ferme le marché de l'entreprise.

On continue à Cupertino la mise au point du SuperMac, extensible (nom de code Milwaukee, première sortie, peut être en septembre), du nouvel Apple // et de leurs périphériques communs (le nouvel Apple // se verrait ainsi doté lui aussi d'une interface SCSI). Du côté du monde extérieur, Macintosh devrait très bientôt pouvoir fonctionner sous Unix, et on reparle aux États-Unis d'une utilisation du MS Dos qui ne passerait pas par l'intermédiaire d'un "MacCharlie" (cette extension géniale transformant le Mac en un vulgaire compatible IBM).

Voilà pour l'avenir. Le présent se conjugue pour le Macintosh grâce à de nouveaux programmes. Apple sait que le développement du logiciel sera désormais capital et a décidé d'en confier la responsabilité à Del Yocam, son directeur général. Trois axes, ce mois-ci, pour les logiciels nouveaux : la boulimie avec le Servant, la communication et l'édition électronique.

## Boulimique

C'est **The Servant** (le serviteur), un domestique doré sur tranches développé par **Andy Hertzfeld** ; après son génial numériseur d'images Thunderscan, il récidive avec ce nouveau logiciel "intégrateur", qui permet d'utiliser simultanément plusieurs logiciels et de repasser de l'un à l'autre. Ça vous rappelle quelque chose ? Et oui, The Servant ressemble à un Switcher mâtiné de Finder. Avec quelques détails qui comptent : ce logiciel (reservé au Macintosh Plus) utilise une mémoire cache de 700Ko. Il permet d'utiliser comme fond du bureau des images numérisées, ou n'importe quel document MacPaint. Quand on ouvre un programme, le bureau ne disparaît pas : l'application s'ouvre simplement dans une fenêtre. On peut garder quatre fenêtres ouvertes simultanément.

The Servant permet de lancer des tâches de fond, c'est-à-dire de laisser l'ordinateur s'occuper en arrière plan d'imprimer, ou de recevoir un fichier par modem, tout en conservant la possibilité de travailler. The Servant est encore un prototype qui devrait être lancé aux États-Unis en septembre.

## Communication

Échanger des données entre un Mac et un IBM ? Facile avec **MacLink de Dataviz**. Ce logiciel comprend deux disquettes : l'une pour le Mac et l'autre pour l'IBM (avec au moins 192Ko et deux lecteurs) ainsi qu'un câble. On relie les deux machines par les sorties séries (ou un modem). Là où les choses se corsent, c'est quand ce logiciel permet de traduire les fichiers pour qu'ils soient utilisables sur l'une ou l'autre machine. Ainsi, un document écrit avec Macwrite peut être converti au format de Displaywrite Wordstar ou Multimate, et être traité sur l'IBM. Bien évidemment, l'inverse est également possible. On peut aussi récupérer une feuille de calculs construite avec Excel, Jazz ou Multiplan et la retravailler avec Lotus 1-2-3, Symphony, etc. Prix (avec un câble) : 155 dollars.

Pour les réseaux locaux, voici un logiciel transformant un Macintosh en serveur : **MacServe d'Infosphere**. Il s'utilise avec le réseau Appletalk et un disque dur Apple, ou un Hyperdrive. Un utilisateur du réseau peut aussi utiliser les bases de données multi-utilisateurs (comme Omnis 3 ou 4e Dimension), et se servir du disque dur comme "spooler", pour imprimer sans immobiliser l'ordinateur. Prix : 250 dollars.

Voici encore les réseaux Omninet (distribués en France par Access) sur lesquels on peut connecter simultanément des Macintosh et des IBM PC, (en tout 62 micro-ordinateurs sur une distance de 1200 mètres) utilisant les disques durs Corvus Omnidrive. Possibilités, là aussi, de partager les mêmes ressources : fichiers, imprimantes, disques durs. Une version Omnitalk permet d'utiliser les disques durs en multi-utilisateurs sur un réseau Appletalk.

## Édition électronique

C'est vraiment la dernière folie à la mode : aux États-Unis, les entreprises succombent aux charmes de l'édition électronique. Certains utilisateurs commencent d'ailleurs aussi à en éprouver les déconvenues : défaut d'encrage notamment sur les imprimantes laser. Il s'agirait d'un problème dû aux cartouches de recharge fabriquées par le japonais Canon. En attendant qu'il soit réglé, un programmes permet de créer des polices de caractères pour l'imprimante Laser, c'est **Fontographer d'Altsys**, permettant de dessiner des polices, de les manipuler (rotations, ombrages, etc.). Un autre logiciel créé par **Manhattan Graphics**, l'éditeur de Ready Set Go, et baptisé **Desk Design** est tout simplement constitué de "templates", c'est-à-dire de maquettes déja prêtes pour éditer une lettre d'information, une brochure... Bref, c'est du clés en mains (préformaté, avec caractères selectionnés) où il ne reste plus qu'à ajouter le texte. Prix : 70 dollars. **BIP** qui distribue en France Ready Set Go en réalisera t-il une VF ?

Plusieurs revues se sont créées, outre-Atlantique, exclusivement vouées à l'édition électronique. Parmi elles : **Publish**, édité par le groupe publiant déjà la revue **Macworld**. On trouve même des leçons d'édition sur vidéo-cassettes (malheureusement illisibles à moins de disposer d'un magnétoscope tri-standard !).

## Des logiciels pour le Mac

La décision d'Apple de ne plus livrer systématiquement Mac Write avec le Macintosh Plus, pour susciter la création de logiciels de traitement de texte, commence à porter ses fruits. Voici, en provenance de Grande Bretagne, **MacAuthor, d'Icon Technology**, un traitement de texte bourré d'idées ingénieuses

permettant notamment d'écrire en colonnes (comme un logiciel d'édition électronique) de préparer des feuilles de style (comme Word), auquel ses premiers utilisateurs ne reprochent qu'une lenteur sensible surtout sur le 512Ko.

**ALS** se prépare à lancer **Word Handler**, un traitement de texte permettant notamment d'utiliser plusieurs fenêtres, qui sera vendu 80 dollars (et 30 à ceux qui renverraient leur disquette originale MacWrite avant juin). **LM Software** va vendre **Macspec**, traitement de texte destiné à ceux qui veulent établir des spécifications, des devis et des notices techniques (avec table des matières automatique), **SSI** prépare une version pour le Mac de son traitement de texte **WordPerfect**, et **Haba** va vendre 100 dollars **Haba Word**, qui permet, comme **MacAuthor** l'écriture sur plusieurs colonnes, avec affichage à l'écran.
Du côté de **Microsoft**, on met la dernière main à **Works**, un programme qui viendra bientôt faire la pige à **Jazz** : télécommunications, base de données, tableurs, et surtout un traitement de texte qui fait déjà saliver avec notamment des possibilités graphiques (dessin, tracé de lignes, etc.) intégrées. Ce programme là devrait être ultra-rapide, car résidant en RAM.
Pour faire bonne impression, **Silicon Press** de **Silicon Beach Software** permet d'imprimer des étiquettes, des cartes en utilisant les données extraites de la plupart des fichiers. En couleurs avec la nouvelle Imagewriter II.

## Bases de données

Pendant un temps, le Mac a été le roi de la base de données : une nouvelle chaque semaine, structure de ses fichiers oblige. Et les français avec **4ème dimension** et **MacBase** n'ont pas été les derniers à donner. Les cadences se sont quelque peu ralenties chez nous mais, aux États-Unis, la gestion de fichier fait toujours florès. Avec le plus simple **Record Holder** de **Software discoveries**. Une petite gestion de fichier facile et rapide, permettant de disposer de champs de 32Ko, de calculer, avec sauvegarde automatique. Ce programme ne permet toutefois pas de stocker des images ou des dessins. Prix : 50 dollars.
Pour le double de ce prix, **MacRelax** d'**Haba** est une base de données relationnelle puissante et agréable d'emploi. Enfin, **Interlace** de **Singular Software** est également, pour 95 dollars, une base de données relationnelle, en même temps qu'un générateur d'applications. On y trace visuellement les liens entre les champs des fichiers, exactement comme dans **4ème dimension**. Cette gestion de fichiers s'adresse surtout à ceux qui manipulent des chiffres et des calculs complexes.

## CAO et graphiques

Le Macintosh accède aux possibilités des stations de conception graphique assistée par ordinateur. Avec **E-Z Draft** de **Bridgeport**. Ce logiciel permet de réaliser des projections avec vue isométrique ou orthogonale, de prélever et d'utiliser les dessins stockés sur des ordinateurs centraux, de créer des notices techniques, tout en utilisant toutes les possibilités du Macintosh et notamment en important des textes de Macwrite et des dessins de MacDraw. Prix : 2495 dollars. Seulement...
**Easy 3D** d'**Enabling Technologies**, importé par **Infotique** est aussi un logiciel graphique permettant de "modeler des surfaces cachées" pour artistes, architectes, ingénieurs, etc. Aussi facile qu'un jeu de pâte à modeler électronique. Toutes les perspectives tridimensionnelles sont possibles, ainsi que les éclairages différents, etc.
**The Graphics Magician** de **Polarware** avait permis aux possesseurs d'Apple // de s'initier à l'animation, à la décomposition des mouvements et de créer des séquences animées. Le voici disponible pour le Macintosh utilisant toutes ses facilités.

## Utilitaires

A noter une version 1.1 de **SideKick**, les accessoires de bureau de **Borland**, qui fonctionnent avec le Macintosh Plus. A remarquer encore une version de **Copy II Plus** capable elle aussi d'utiliser le Macintosh Plus et ses disquettes 800K...
Enfin, **HD Util** de **FWB Software** peut recopier sur un disque dur des programmes théoriquement protégés. Interceptant la routine demandant l'introduction de l'original, il permet de se passer de celui-ci. Prix : 90 dollars. A remarquer, qu'il est nécessaire d'utiliser un "patch" particulier à chaque programme installé.

## Matériel

Du "hard" pour le Mac. Avec d'abord des disques de **Warp Nine**. Un disque dur externe d'une capacité de 20 Méga-octets pour 795 dollars, un disque interne à la façon de l'**Hyperdrive** de 20 Mégas aussi pour 895 dollars. Et un lecteur de disquettes 800Ko pour 239 dollars.
**Apple** se prépare à sortir une version adaptée à l'interface SCSI de son disque dur au moment où ceux-ci commencent à baisser. En France aussi où **P-Ingénierie** a baissé le prix des "Hyperdrive" (15400F HT pour le 10 Mégas et 18900F pour le 20 Mégas). Le même P-Ingénierie importe **Mac Palette de Pixel Software**.

Un ensemble composé d'un logiciel, d'un moniteur et d'une modification du Macintosh qui permet de transformer le Mac en station graphique professionnelle en couleur de haute qualité. (résolution de 768 x 576 pixels, 256 couleurs simultanées dans un choix de 16 millions). Pour quelques 150 000 Francs (Mac compris), ce système permet d'obtenir sur l'écran, ou sur des films photos ou des transparents, des images couleurs de haute qualité. Dommage que son prix le réserve aux "pros".
A remarquer, aussi, un buffer pour imprimante, le **MacBuffer** d'**Ergotron**. On peut y installer de 256Ko à 1 méga de mémoire, et pouvoir ainsi utiliser le Mac pendant que l'imprimante imprime. A noter que ce buffer est parfaitement dessiné pour s'installer à côté de l'Imagewriter I, mais fonctionne aussi avec la nouvelle. Il permet accessoirement à deux Mac de se partager une imprimante.
Enfin, **MacMemory** dont les produits sont importés par **CCAM** a déjà étudié une augmentation de mémoire pour le Macplus. Il s'agit d'une barrette qui s'installe sur l'emplacement disponible dans le nouveau Mac et porte instantanément sa mémoire disponible à 2 Mégas.

## Et l'Apple // ?

A l'exception d'une nouvelle version d'**Appleworks** (la 1.3) tirant parti du lecteur de 800Ko et des cartes d'augmentation de mémoire, les nouveautés portent surtout sur le matériel.
Une société américaine ("**The Engineering Department**") a développé une carte permettant au //e d'utiliser les logiciels de l'IBM. Pour 500 dollars, auxquels il faut ajouter un lecteur compatible avec le PC et un clavier type IBM. Dommage qu'avec la chute des prix des compatibles venus de Taiwan, l'ensemble revienne désormais quasiment au prix d'un "clone" complet.
**Video Technology** a tenté de vendre aux Etats Unis son **Laser 128**, une réplique de l'Apple //c nanti en prime d'un pavé numérique et d'une connection pour relier un supports de slots d'extension pour 395 dollars. Si les douanes américaines n'y ont rien trouvé à redire, Apple a prévenu que l'appareil relevait de la violation du copyright et demandé aux tribunaux d'en interdire la vente.

## Freeware

Ce sont ces logiciels disponibles gratuitement ou moyennant une petite redevance, qui comprennent d'utiles accessoires de bureau, des utilitaires, des jeux, etc. Aux Etats-Unis, on peut les télécharger facilement. En France la pratique commence à se répandre. Les abonnés de **Calvados** peuvent d'ores et déjà, à condition d'être muni d'un modem, d'un logiciel de télécommunications évolué (comme MacTel 2), et d'un utilitaire de conversion (comme Binhex) se procurer un éditeur de ressources comme Resedit, un installateur de touches de fonction, comme FK installer, etc.
En Grande Bretagne, le serveur Mactel auquel on peut s'abonner moyennant une cotisation peu élevée de 10 Livres sterling, offre les mêmes services avec toute une foule de logiciels récents et intéressants (notamment la version 4.6 du Switcher, etc.).
On peut se brancher en 1200/75 bauds sur ce serveur en appelant le 602 817 696 à Nottingham (de France composer d'abord le 19 puis le 44). Si les lecteurs de Pom's connaissent d'autres serveurs réservés au Mac, qu'ils nous en indiquent les coordonnées et celles ci seront publiées ici.

## Adresses

**Dataviz** - 16 Winfield St, Norwalk CT06855
**Infosphere** - 4730S.W.Macadam Av. Portland, Oregon 97201
**Access** - 10, rue Ampère Prolongée Parc Technologique de Bois d'Arcy - 78390 Bois d'Arcy
**Altsys** - PO Box 865410 Plano TX 75086
**Publish** - PO Box 51966 Boulder Colorado 80321
**Silicon Beach Software** - 9580 Black Mountain Road Suite PO Box 261430 San Diego CA92126
**Software discoveries** - 99 Crestwood Rd Tolland CT 06084
**Haba** - 6711 Valjean Av. Van Nuys CA 91406
**Singular Software** - 5888 Castano Drive San Jose CA 95129
**Bridgeport Machines** - Bridgeport CT
**Infotique** - 20 et 24, rue de l'Hôtel Dieu 95300 Pontoise - Tel.: 30 73 14 15
**FWB Software** - 2040 Polk St #215 San Francisco, CA 94109
**Warp Nine** - 3216 Humbolt Av. Minneapolis MN 55408
**P-Ingénierie** - 226 Bd Raspail 75014 Paris - Tel. 43 21 93 36
**Ergotron PO** - Box 17013 Mpls , MN 55417
**CCAM** - 95, rue Lafayette - 75010 Paris - Tel. : 47 80 22 23
**Calvados** - BP 21-07 75327 Paris Cedex 07
**Mactel** - 15 Elmtree av, West Bridgford, Nottingham NG2 7JV Grande Bretagne

*Alexandre Duback*

# Bibliographie

**Entreprendre et gérer sur Macintosh (modèles Multiplan)**, de Bill Bono et Ben Kalkis – Cedic-Nathan – 303 pages – 225 F – Traduction.

Ces deux ouvrages regroupent des modèles de gestion prêts à utiliser. Le travail est bien fait et la maquette bien conçue ; le seul problème, comme pour tous les ouvrages du genre, est le suivant : les problèmes propres à une entreprise peuvent-ils facilement être adaptés à un tableau tout prêt, ou est-il plus rapide de construire son propre tableau en partant de rien ? Je penche pour la seconde option. De toute façon, il reste l'intérêt pédagogique des exemples.

---

**50 modèles Multiplan pour gérer sur Apple II et IBM PC**, de Fabienne et Philippe Gysel – Editions du PSI – 196 pages – 130 FF.

Mêmes remarques que pour les deux ouvrages précédents. A noter cependant que certains des principes essentiels de la bonne programmation sur tableurs présentés dans les ouvrages de H. Thiriez ne sont pas appliqués : ainsi, la somme d'une série de valeurs n'est-elle jamais "englobée", ce qui rend les formules nettement moins adaptables. L'englobage consiste à faire la somme depuis une cellule avant un bloc (si elle ne contient pas de valeur) jusqu'à une cellule après ce bloc (idem) au lieu de l'effectuer seulement entre les deux bornes : on peut ainsi insérer ou détruire à liberté dans le bloc sans remettre en cause la validité de la formule.

---

**Bases de données sur Macintosh**, de Frédéric Joutel et Eric Tenin – Edimicro – 149 pages – 88 FF.

Faites le calcul : 9 "bases de données" en moins de 130 pages – introduction et conclusion exclues – cela fait moins de 15 pages par programme, copies d'écran comprises. Certains logiciels sont présentés dans leur version US, alors que la version française existe. Il n'y a pas de tableau récapitulatif comparant les programmes en fonction d'un certain nombre de critères, pas d'analyse critique sérieuse,... Bof.

**Utiliser les bases de données avec dBase II et dBase III**, de Claude Frasson – Editions d'Organisation – 250 pages.

Voici en revanche un livre bien fait, pédagogique, et avec de nombreux exemples de programmes. Un seul regret : le côté spartiate du maquettage, ce qui devient rare aujourd'hui, et la couverture à spirale de la collection qui, si elle facilite la lecture à plat sur une table, ne simplifie pas le rangement dans une bibliothèque.

---

**Utiliser Wordstar**, de Maureen A. Culleeney – Editions d'Organisation – 271 pages – Traduction.

Mêmes remarques que ci-dessus en ce qui concerne le maquettage et la couverture. Heureusement, dans ce livre, de nombreux petits dessins et des tableaux encadrés rendent la lecture plus agréable. Un livre à conseiller pour un apprentissage approfondi de Wordstar.

**Les 19, 20, 21, et 22 juin 1986, Pom's sera à**

**Apple Expo**

**au Parc de la Villette (Grande Halle), porte de Pantin à Paris.**
**Nous vous attendons au stand B-10.**

**N'oubliez pas de présenter ce numéro de Pom's : vous entrerez à demi-tarif...**

Un collaborateur de Pom's vend à l'état neuf (sous emballage), un lecteur 140Ko //c (2000,00 F), un moniteur //c (1300,00 F), un stand //c (250,00 F), une housse //c (375,00 F), un joystick //e-//c (375,00 F) et un manuel de référence //e (250,00 F). Ecrire à la rédaction qui transmettra.

# Courrier des Lecteurs

*J'aimerais poser quelques questions au sujet du compilateur TASC (compilateur Applesoft distribué par Microsoft) :*

- *comment est organisé le rangement des chaînes de caractères ?*
- *existe-t-il une routine de nettoyage mémoire rapide qui fonctionne avec TASC ?*
- *peut-on réécrire les chaînes de caractères au même emplacement mémoire si elles ont la même longueur et le même nom ?*

*Dominique OTTELLO, 75017 Paris*

Il est impossible de détailler ici le fonctionnement de TASC et en particulier du fichier RUNTIME utilisé pendant l'exécution d'un programme compilé.
Schématiquement, on peut préciser les points suivants :

- La méthode de rangement des variables est comparable à celle de l'Applesoft : les chaînes de caractères sont stockées en descendant à partir de la valeur du HIMEM.
- Chaque utilisation de la même variable fait appel à une même routine, générée par le compilateur, qui permet de déterminer l'adresse du descripteur de la variable. Ce descripteur pointe lui-même vers la valeur courante de la variable précédée de sa longueur.

Ainsi, par exemple, le programme :

```
10  INPUT A$ : PRINT A$
```

après compilation, contient à partir de l'adresse $17B0, des instructions qui appellent, en des points d'entrée différents, une routine située en $17DC qui gèrent tous les accès à la variable A$ en précisant que son descripteur est en $17E6.

Après une exécution où la valeur de A$ est une chaîne de 40 "A", le descripteur pointe vers l'adresse $95D6. Ici, on y trouve $28 (la longueur de la variable), suivi de 40 octets contenant $41 ("A").

- Contrairement à l'Applesoft, TASC ne duplique pas les chaînes identiques ; l'instruction :

```
20 B$ = A$
```

crée un descripteur qui pointe également vers l'adresse $95D6.

- Une routine de nettoyage mémoire est intégrée dans TASC et appelée sous les mêmes conditions que l'Applesoft. Elle raffraîchit l'occupation des chaînes et réorganise les descripteurs.

---

*Ayant découvert votre revue par le numéro 21, je voudrais savoir comment sauvegarder, après les avoir saisi au clavier, les fichiers SET et FORMES4 du programme EDIGRAPH.*

*J. BUTTANI, 68000 Colmar*

Deux instructions suffisent (elles peuvent être déduites des pages 16-17) :

```
BSAVE SET, A$3000, L$F00
BSAVE FORMES4, A$6000, L$2
D5
```

---

*Ayant rentré le programme HGR DUMP pour Epson (Pom's 17), j'ai obtenu des résultat bizarres qui ne correspondent pas à ce que j'attendais.*

*F. Cabanat, 75011 Paris*

Par omission, l'article ne précisait pas que le programme était destiné à fonctionner avec une carte d'interface parallèle compatible avec celle diffusée à l'origine par Apple-Seedrin. En particulier, il ne peut donc fonctionner sans modification avec une carte série.

Si vous connaissez l'assembleur, les lignes 536 à 538 du programme source contiennent les instructions d'appel à la carte parallèle : boucle jusqu'à ce que l'imprimante soit prête, puis envoi de 8 bits simultanément.

---

*M'étant remis au Pascal, j'ai voulu utiliser les programmes en Pascal fournis sur les disquettes Pom's, après les avoir transférés à l'aide du programme BASPAC. Voici ce que j'ai constaté (mon équipement est classique : Apple II+ et carte langage) :*

- *si on part d'une disquette Pascal qui contient déjà un ou plusieurs fichiers, le programme que l'on trouve après transfert n'a rien à voir avec le programme original mais semble être une partie tronquée d'un des programmes existants ;*
- *en partant d'une disquette fraîchement initialisée et ne contenant aucun fichier, la date du directory est remplacée par des "???", ce qui n'est d'ailleurs pas gênant ;*
- *je n'ai pas réussi à transférer le programme POINTEURS du Pom's 18 ; je n'ai obtenu que des suites incohérentes d'octets, dont beaucoup de caractères de contrôle.*

*André Pettelat, 75011 Paris*

Nous avons également constaté ce problème avec BASPAC lors de certaines conversions de fichiers, bien qu'il ne soit pas systématique. Lorsque BASPAC parvient à transférer un fichier avec succès sur une disquette déjà utilisée, la date du fichier est normalement la même que celle du fichier précédent du directory; en revanche, sur une disquette

fraîchement initialisée, la date est encodée de manière à donner un affichage "???" au catalogue.

En cas de problème lors du transfert, il est donc préférable de le recommencer en utilisant une disquette Pascal dont le directory est vide.

Enfin, le fichier POINTEURS contenu sur la disquette Pom's 18 contenait par erreur le code objet et non le source. Vous le trouverez sur la disquette Pom's de ce numéro.

---

*Parmi vos anciens numéros, en existe-t-il un répondant au problème suivant : d'un côté un programme en Pascal, de l'autre des fichiers de données sur disquettes DOS 3.3.*

*Comment puis-je convertir les fichiers pour qu'ils puissent être traités correctement par le programme en Pascal ?*

*Jean Claude Decret, 33600 Pessac*

Le programme BASPAC contenu sur toutes les disquettes des numéros de Pom's où se trouvent des programmes Pascal peut vous aider.

Il faut néanmoins avoir, au préalable, converti vos fichiers de données (probablement sous format texte), en fichier de format binaire avec le bit de poids fort de chaque octet à zéro. Le programme Applesoft suivant en est une ébauche :

```
10 D$ = CHR$(4) : A = 4096
20 F1$ = "FICHIER.TEXTE" :
   F2$ = "FICHIER.BIN"
30 PRINT D$ "OPEN" F1$
40 ONERR GOTO 110
50 PRINT D$ "READ" F1$
60 INPUT A$
65 IF A$ = "" THEN 95
70 FOR I = 1 TO LEN(A$)
75 B = ASC(MID$(A$,I,1)) :
   IF B > 128 THEN B = B -
   128
80 POKE A, B
90 A = A + 1 : NEXT
95 POKE A, 13 : A = A + 1
100 GOTO 60
110 PRINT D$ "CLOSE" F1$
120 PRINT D$ "BSAVE" F2$ "
   ,A4096,L" A - 4096
```

qu'il faudra adapter si vos fichiers texte sont trop gros pour être contenus dans un seul fichier binaire, s'ils contiennent des virgules ou sont à accès direct.

---

Cher lecteur du Minimic sur Rance, votre lettre relative à un programme Pascal pour impression mathématique ne mentionne ni votre nom, ni le nom de la rue...
Pouvez-vous prendre contact avec la rédaction ?

---

*Un lecteur (courrier des lecteurs du 21) cherchait à modifier la version américaine d'AppleWriter II pour obtenir sur son imprimante le caractère «ç». Il suffit de modifier 5 octets pour réaliser le soulignement avec le caractère «'».*
*Ces 5 octets à modifier font partie du fichier OBJ.APWRT situés aux emplacements suivants :*

*Piste 12,*
*secteur 0B, octet AC ;*
*secteur 0A, octet 12 ;*
*secteur 09, octet A9 ;*
*secteur 09, octet DC ;*
*secteur 08, octet 01.*

*vous devez à chaque fois trouver 'C9 DC'.*
*Il faudra remplacer ces valeurs par 'C9 E0' avec votre éditeur de secteurs favori.*

*J.-M. GUILBERT, Vaujours*

Merci pour cette solution. Doit-on ajouter qu'il est indispensable d'apporter ces modification sur une copie d'AppleWriter ?

---

*Plusieurs lecteurs utilisant COGO nous ont demandé que faire pour y apporter des modifications.*

Modifier COGO est possible mais pénible du fait de la lenteur de Pascal et de la faible capacité des disquettes.

- Il faut 128Ko. Le //e avec carte Chat Mauve permet de travailler sur COGO 560, un //e avec une autre carte ou un //c ne permettent de travailler qu'avec COGO 280.
- Vous devez utiliser le système APPLE.PASCAL 1.2, 64 Ko ou 128 Ko, peu importe.
- Vous devrez travailler sur une copie de APPLE2 et deux copies de COGO (560 ou 280 c'est selon). L'une jouera le rôle d'APPLE1, l'autre accueillera les versions modifiées.
- En bootant votre disquette APPLE1 et utilisant le FILER :
  . Supprimer de APPLE2: les fichiers SYSTEM. ASSEMBLER, 6500. OPCODES, 6500.ERROR et CRUNCHER.
  . Copier tous les fichiers TEXT de COGO: sur APPLE2: et effacer de COGO: tous les fichiers de COGO (.TEXT, .CODE et exemple)
  . Copier sur COGO: les fichiers suivant de APPLE1: SYSTEM.SYNTAX, SYSTEM.EDITOR, SYSTEM.FILER.
  . Cruncher COGO: et changer son nom en APPLE1: par exemple.
- APPLE1: contient le système APPLE 1.2 128 Ko utilisé et doit être maintenant booté, tandis que APPLE2: doit être mis dans le lecteur 2.
- Il n'y a plus qu'à éditer et compiler COGO, puis le sauver avec le FILER sur la copie de COGO: destinée à accueillir les versions modifiées.

A noter qu'il ne faut pas oublier, avant de compiler COGO 560/280 :

. soit de transférer EDIT COGO.CODE dans SYSTEM. LIBRARY en utilisant le programme LIBRARY de APPLE2 ; en effet, COGO utilise l'UNIT EDITION, qui est créé par EDIT- COGO.
. soit de mettre dans COGO 560/280.TEXT :
USES...,..., (*APPLE2:EDIT COGO.CODE*) EDITION;

# Bon de commande

## Disquettes

| | | | | |
|---|---|---|---|---|
| HAIFA source | (cf. Pom's n° 5) | ..... | à 60,00 F | ..... |
| H-BASIC | (cf. Pom's n° 8) | ..... | à 150,00 F | ..... |
| MUSIC | (cf. Pom's n° 10) | ..... | à 80,00 F | ..... |
| DISK-MANAGER | (cf. Pom's n° 11) | ..... | à 450,00 F | ..... |
| BASICIUM | (cf. Pom's n° 13) | ..... | à 150,00 F | ..... |
| E.P.E. 5.0 | (cf. Pom's n° 23) | ..... | à 200,00 F | ..... |
| Échange E.P.E. 5.0 | (cf. Pom's n° 23) | ..... | à 80,00 F | ..... |
| PASCAL | (cf. Pom's n° 15) | ..... | à 80,00 F | ..... |
| MAX (Moniteur étendu) | (cf. Pom's n° 18) | ..... | à 150,00 F | ..... |
| DOMINOS | (cf. Pom's n° 19) | ..... | à 80,00 F | ..... |
| MACASTUCES | (cf. Pom's n° 21) | ..... | à 200,00 F | ..... |
| P-FORMAT ][, P-POLICE ][ | (cf. Pom's n° 21) | ..... | à 200,00 F | ..... |
| COGO | (cf. Pom's n° 21) | ..... | à 150,00 F | ..... |

## Recueils

| | | | |
|---|---|---|---|
| N°1, recueil des revues 1 à 4 | ..... | à 140,00 F | ..... |
| Disquettes d'accompagnement 1 à 4 | ..... | à 200,00 F | ..... |
| N°2, recueil des revues 5 à 8 | ..... | à 140,00 F | ..... |
| Disquettes d'accompagnement 5 à 8 | ..... | à 200,00 F | ..... |
| N°3, recueil des revues 9 à 12 | ..... | à 140,00 F | ..... |
| Disquettes d'accompagnement 9 à 12 | ..... | à 200,00 F | ..... |

## Revues, disquettes

| | | | |
|---|---|---|---|
| Revues 4 7 8 *(n° 9 épuisé)* | ..... | à 35,00 F | ..... |
| Revues 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 | ..... | à 40,00 F | ..... |
| Disquettes Apple II, //e, //c<br>1/2 3 4 5 6 7 8 9 10 11 12<br>13 14 15 16 17 18 19 20 21 22 23 24 | ..... | à 60,00 F | ..... |
| Disquettes Macintosh | | | |
| 14/15/16 groupées | ..... | à 150,00 F | ..... |
| 17 18 19 20 21 22 23 24 | ..... | à 80,00 F | ..... |
| Mac 'A' | ..... | à 80,00 F | ..... |

## Abonnements

Pour 6 numéros à partir du n° ☐

| | | | |
|---|---|---|---|
| Abonnement à la revue seule | ..... | à 200,00 F | ..... |
| Abonnement revue + disquettes Apple II, //e, //c | ..... | à 500,00 F | ..... |
| Abonnement revue + disquettes Macintosh | ..... | à 600,00 F | ..... |

**Total TTC :** _____

Supplément avion hors CEE : 15,00F par numéro et/ou disquette : _____

**Montant du règlement :** _____

*Envoyez ce bon et votre règlement à :* EDITIONS MEV, 64 rue des Chantiers 78000 VERSAILLES

**Nom :** .....

**Adresse :** .....

*Imprimerie Rosay, 94300 Vincennes, Imprimé en France. Dépot légal : mai 1986. N° 3368*

# P.S.I. VISE JUSTE

**P.S.I. EST A APPLE EXPO**
du 19 au 22 JUIN 86

## PROGRAMMER

**C**lefs pour Macintosch **150** FF – **B**asic Microsoft 2.0 sur Macintosh **250** FF – **B**asic + 80 routines sur Apple II **95** FF – **L**es ressources de l'Apple IIc **95** FF – **A**ssembleur de l'Apple **120** FF – **I**ntroduction à ProDOS sur Apple **85** FF – **S**ystème ProDOS sur Apple **190** FF – **P**rogrammation système de l'Apple II **190** FF – **A**pple, modems et serveurs **130** FF – **C**lefs pour l'Apple IIc et IIe 65CO2 **145** FF.

## UTILISER

**M**ac Astuces **150** FF – **M**ultiplan pour Macintosh **110** FF – **L**e livre de Jazz **220** FF – **5**0 modèles Multiplan pour gérer sur Apple et IBM/PC **130** FF – **A**ppleworks au travail **160** FF – **P**hotographie sur Apple et Amstrad **150** FF.

# DES LIVRES POUR

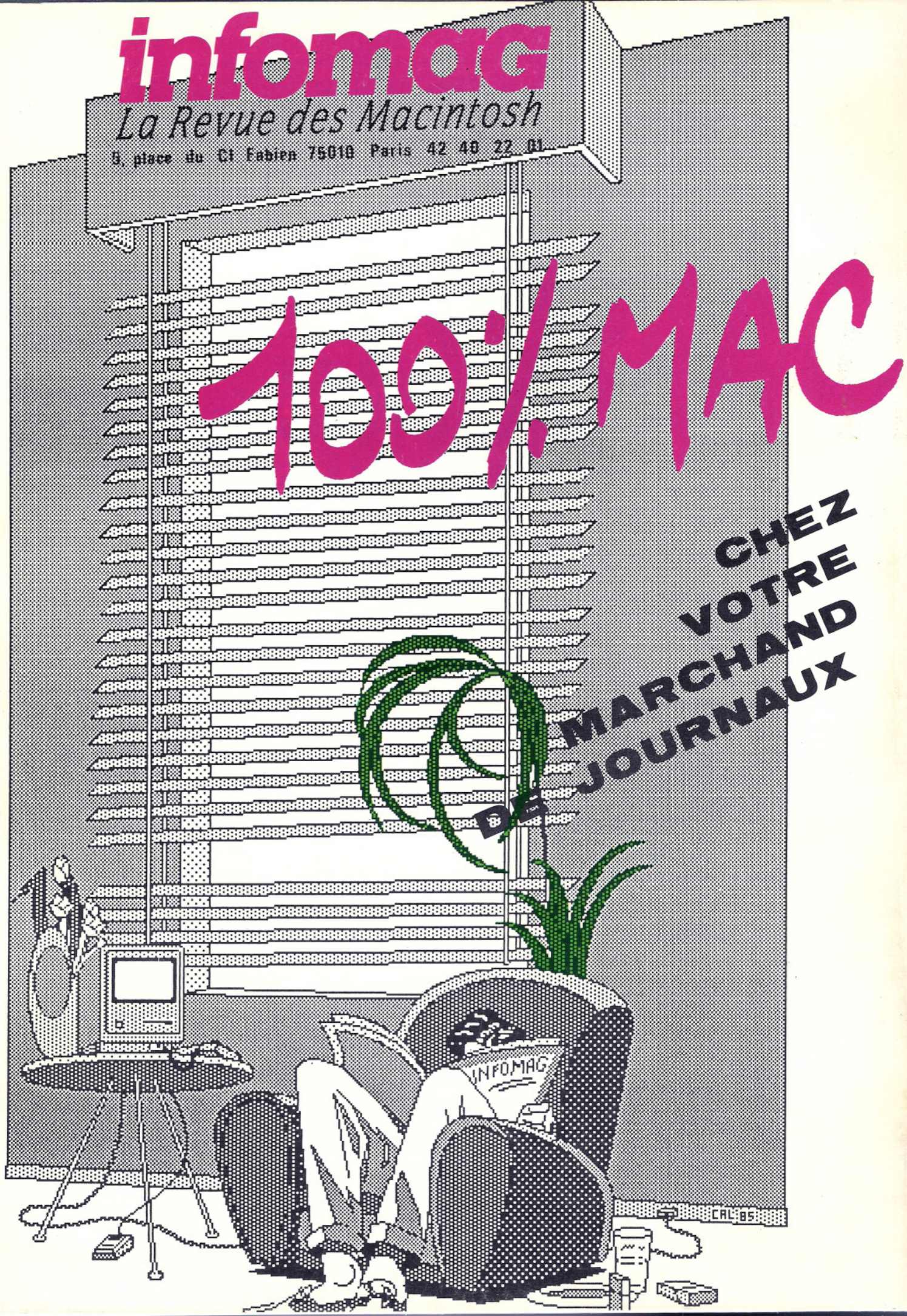
## CRÉER

**P**rogrammation des jeux d'Arcade sur Apple II **140** FF – **A**pple, logique et systèmes experts **120** FF – **C**réation et animation graphique sur Apple **335** FF.

## JOUER

**1**02 programmes pour Apple **120** FF – **S**uper jeux Apple **120** FF.

**DEMANDER LE CATALOGUE GRATUIT** à P.S.I. Diffusion - B.P. 86 - 77402 LAGNY CEDEX


infomag
La Revue des Macintosh
5, place du Cl Fabien 75010 Paris 42 40 22 01
100% MAC
CHEZ VOTRE MARCHAND DE JOURNAUX
INFOMAG
CALOS